ファミリーコンピュータ™
ファミリーベーシック

ぼくらのファミコン入門

ファミリーベーシックでオリジナルゲームに挑戦

新星出版社

任天堂 ファミリーコンピュータ™
ファミリーベーシック™

ぼくらのファミコン入門

山下利秋

ファミリーベーシックでオリジナルゲームに挑戦



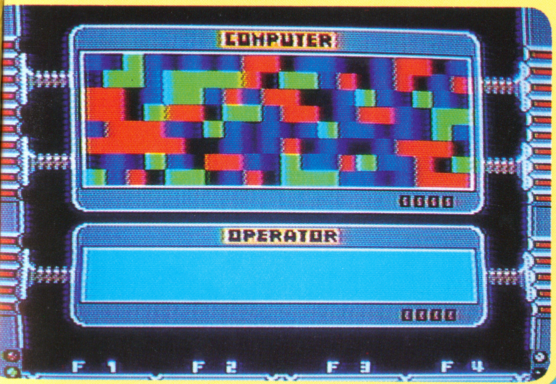
カバーイラスト
田中秀夫
カバーデザイン
峰田順一

きみにだって作れる ファミコンゲーム！

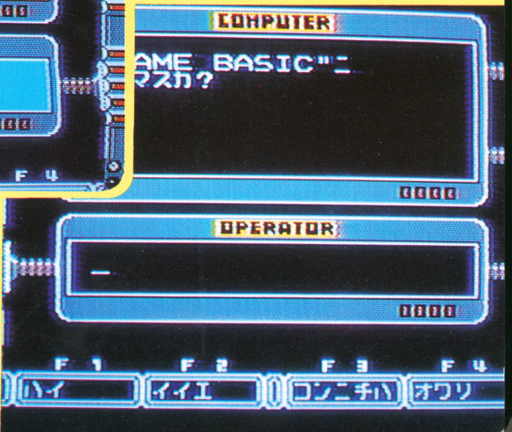


スリリングでエキサイティングで、おもしろくてたまらないファミコンゲーム。どうすればあんなすごいゲームが作れるのだろう？
ところが、きみにも作れるんだ。ファミコンのことばさえ知っていればね。それがファミリーベーシックだ。さぁ、ファミリーベーシックをマスターして、きみだけのゲーム作りにアタック！

ピコピコピコ…ファミコンと通信開始だ！



◀ スイッチON‼テレビの画面にコンピュータの絵が出てきたぞ。ベーシックが動きだした合図だよ。



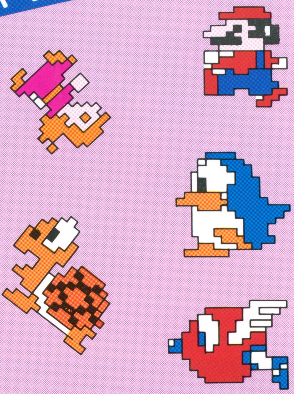
▶ これだ！ゲームのプログラムを作るGAME BASICだ。ここでキーボードの一番上にある[F1]キーを押すんだ。

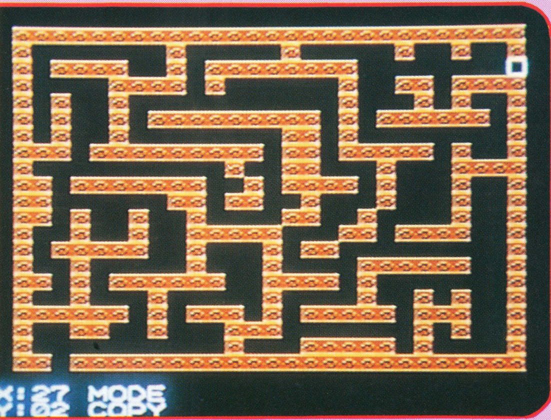
頭をひねってオモシロ画面を作りだせ！
ＢＧグラフィックでゲームの舞台も完ぺきだ



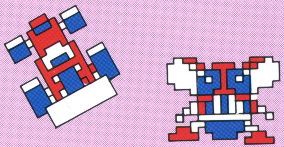
BGグラフィックはブロック遊びみたいだな。104種類もあるキャラクタを画面の上で自由自在に組み合わせてみよう。

キャラクタ大集合！





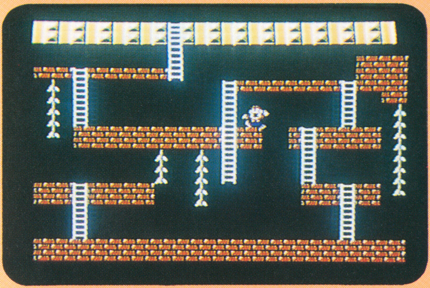
こんなむずかしそうな迷路だって、COPYのテクニックを使えばあっというまだ。ペンペンでも走らせてみれば？

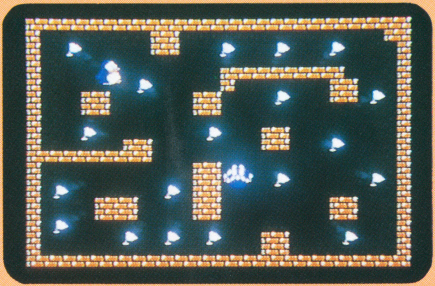




BGグラフィックでコースを作ってプログラムとドッキング。アニメキャラクタ運動会ゲームは180ページにのってるよ。

オモシロゲームはきみのアイデアしだいだ
あとはベーシックとグラフィックをフル作動





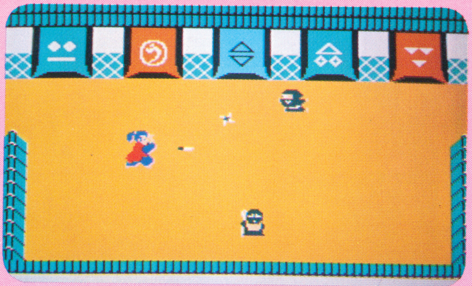
背景の絵を描くのがＢＧグラフィックなら、キャラクタを動かしてゲームのプログラムを作るのがGAME BASICだ。ベーシックの命令はたくさんあるけど、おそれることはない。きみならすぐにマスターしてしまうはずだ。オモシロゲームを作りたいなら、アイデアが勝負だぞ！





きみたちの先輩が作った画面だ。こんなきれいな背景画もやっぱりＢＧグラフィックで作ってあるんだよ。
（画面写真はゲーム「謎の村雨城」　提供＝任天堂）

ビックリしてドッキリしてワクワクワク……
いつか作ってみせるぞ、こんなゲーム！



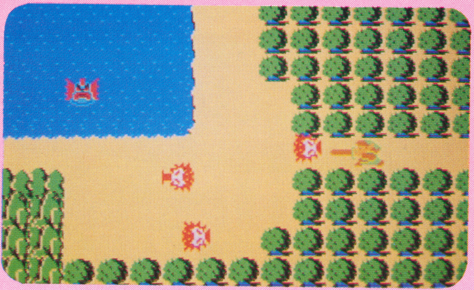
宿　場

そこは忍者たちの巣だった。多勢に無勢、剣士はこの難関を切りぬけられるか…。



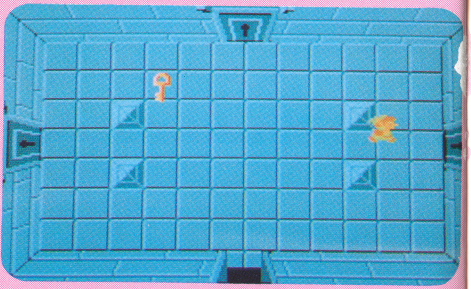
森

木かげにかくれる妖術使いや忍者たち。〝謎の玉〟への道はまだまだ遠い。



泉のほとり

トライフォースの謎を解き明かし、ゼルダ姫を救出しなければならないリンクの行く手には……。



地下迷路

暗い地下の迷宮に入り込んでしまった。不気味だ。いったいどんな魔ものがひそんでいるのだろう。

（画面写真はゲーム「謎の村雨城」「ゼルダの伝説」　提供＝任天堂）

任天堂 ファミリー コンピュータ™
ファミリー ベーシック™

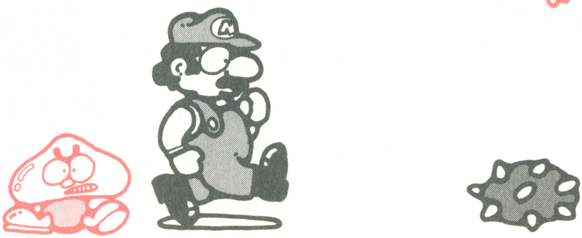
ぼくらのファミコン入門



ファミリーベーシックでオリジナルゲームに挑戦

山下利秋





新星出版社

●はじめに

ベーシックはパソコニストへの第一歩



ワクワク、ドキドキ、エキサイティング！　ファミコンはたしかに、すぐれたゲームマシンだ。ゲームで遊んでいるだけでもじゅうぶんおもしろいのに、どうしてきみは、ファミリーベーシックに興味を持ったのかな？

きみ自身は、まだ気がついていないかもしれない。でも、それはたぶん、きみがたんなるゲームマニアから、パソコニストへの道へ、第一歩を踏み出したしるしだと思うよ。

きみがいままで楽しんできたゲームソフトのかずかずも、結局はだれかが作ったものだ。こんどきみは、その作る立場に立つわけだ。

いきなり、むずかしいゲームを作ることはできないだろう。でも、この本でこれから勉強するベーシックということばは、たくさんあるコンピュータのことばの中でも、もっとも初心者にわかりやすいことばなのだ。

このことばを通じて、いままで知らなかったプログラミングの仕組やゲームの作り方、さらには、奥深いコンピュータの世界の入口をのぞくことができるはずだ。

（著者よりファミコンファンのみなさんへ）

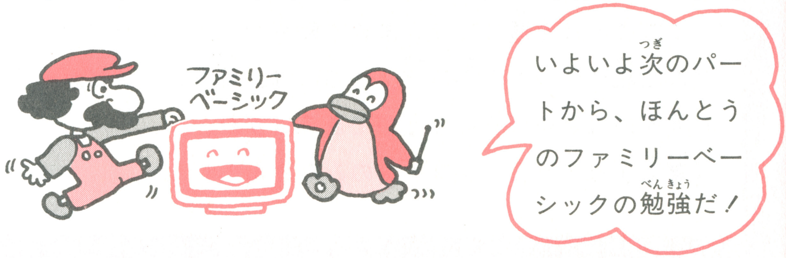
もくじ

♥ファミリーベーシックに入る前に読むページ

PART 0　はじめてのファミコン

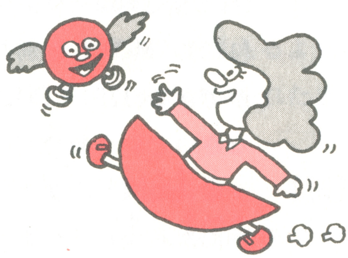
♠GAME BASIC以外の機能を楽しむページ

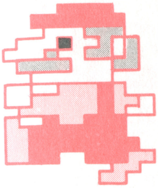
PART 1　はじめてのファミリーベーシック



♣キー操作の初歩から入るページ

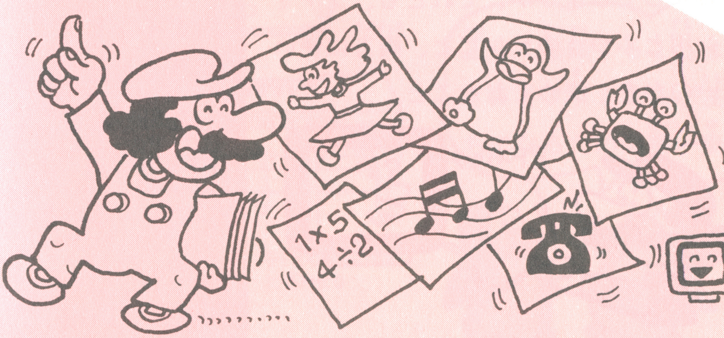
PART 2 ゲームベーシックの初歩

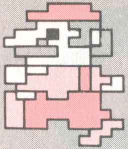




PART 0
はじめてのファミコン

　ようやく手に入ったご自慢のファミコン・マシン。
　さっそくピコピコピコとゲームしたい気もちはよくわかるが、ちょっと待って。
　ゲームするにも、ベーシックするにも、その前にしておかなければならないことが、あるのだよ。





o.1



ファミコンをテレビにつなぐ

テレビも見る？ファミコン専用？ つなぎ方は2とおりあるぞ！

♥きみのファミコンにRFスイッチはあるか？

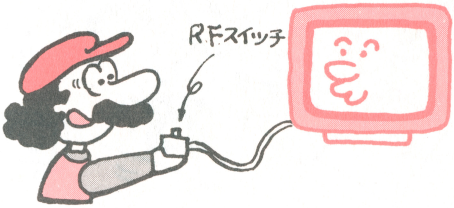
ファミコンは、ふつうのテレビにRFスイッチでつないでカセットをポン。たったそれだけの手順で、すぐにゲームで遊べるのだから、うれしくてたまらない。

でも、テレビとどうやってつなぐのかがわからなければ、遊びたくても遊べない。まず、正しく接続することをおぼえよう。

最初に、ファミコンと接続するテレビを、ファミコン専用のテレビにしてしまうか、それとも、ファミコン用とふつうのテレビと、両方に使いたいのかを、決めておくこと。

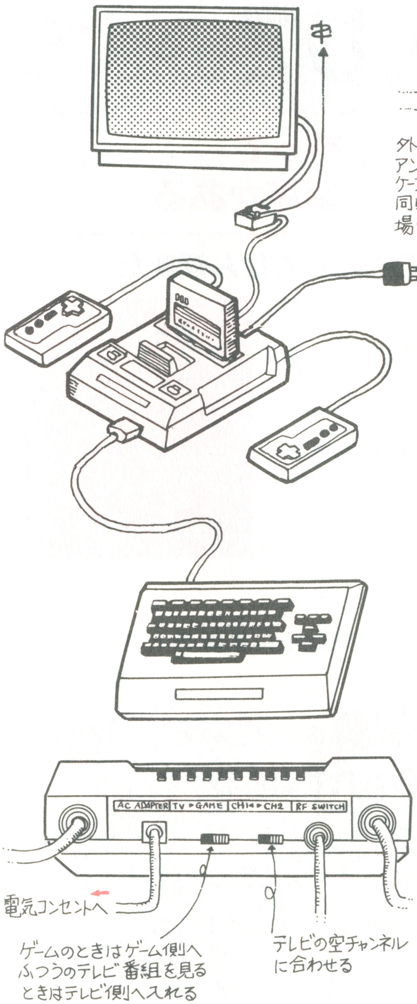
●ポイント

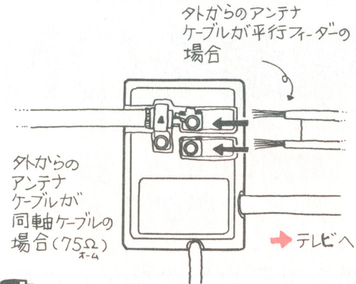
ファミコンを買ったら、かならずRF SWITCHがついているかどうかを、確認すること。

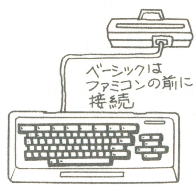


♥ファミコンとテレビをどうやってつなぐか？

■ファミコンとテレビの接続図



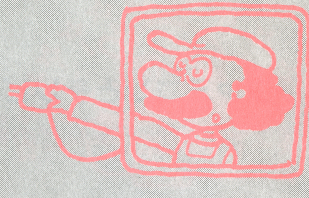








o.2



電源をONしてみよう

電源ON?　ちょっと待って
スイッチの入れ方にも順序がある

♥ファミコンのスイッチONはいちばん最後だ

さあ、テレビと接続できたら、さっそくゲームで遊びたいよね。ファミコン・ファンのきみのことだ。ゲームカセットだけは、しっかり用意していることだろう。カセットを、カセットコネクタに差しこんで………。

> **●ポイント**
>
> カセットは、カチャッという音がするまで、しっかりと差し込むこと。

スイッチON!

おっとっと…。ちょっと待て。やたらに電源を入れてはいけないぞ。

ファミコン本体のスイッチをいれるのは、いちばん最後。

♥スイッチは周辺機器から入れていこう

ファミコンだけじゃなく、どのコンピュータでも、電源をいれるときは、まず周辺機器のスイッチからだ。この順序を守らないと、機械が故障したり、ゲームカセットがだめになることがあるから、ぜったいに守ることだ。

では、スイッチを切るときは、どうかというと………。

こんどは、いれたときと逆の順序できっていく。まず、ファミコン本体のスイッチを切ってから、周辺機器だ。そして、ぜんぶのスイッチを切ったら、最後にカセットをぬくよ。

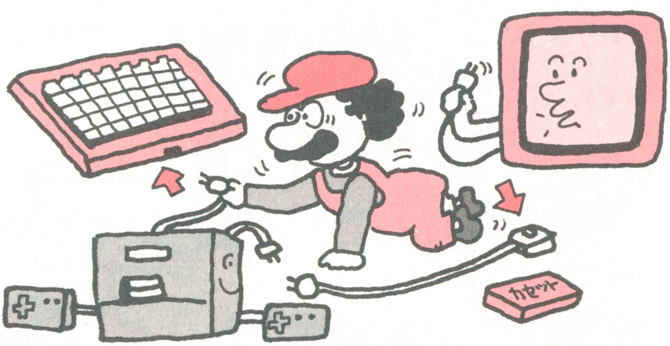
♥周辺機器ってな〜に？

周辺機器というのは、コンピュータの本体に接続して、いろいろな仕事をさせる機械たちのことだ。

> **●ポイント**
> 周辺機器どうしの順番は、どれからいれても、どれからきってもかまわないよ。ここでは、テレビが周辺機器にあたるね。

たとえば、プリンタはプログラムを本体のメモリから紙に打ちだすときに使う機器。カセットレコーダは、本体のメモリに入っているプログラムをカセットテープに保存しておくためのものだ。

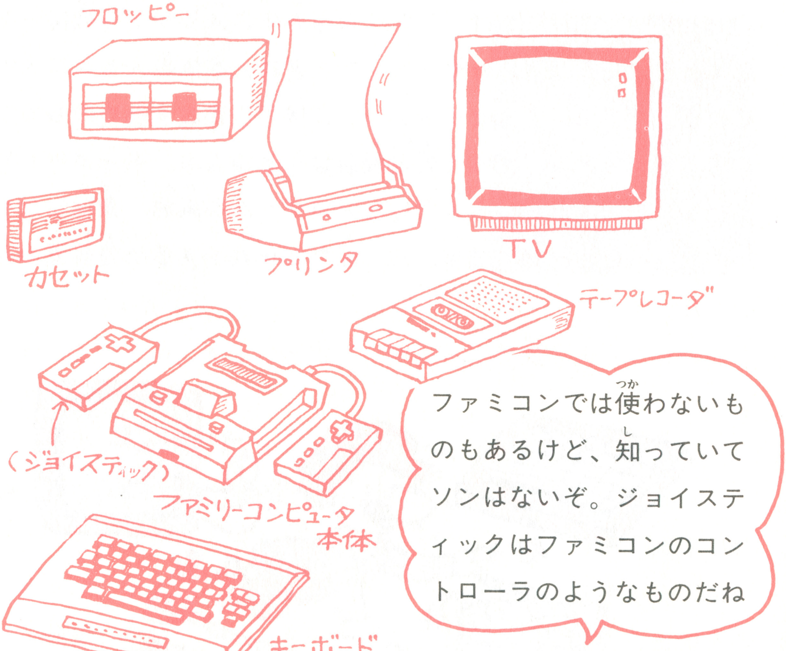
そうそう、テレビだって周辺機器のひとつだよ。キーボードやカセットテープなどから入力したプログラムを画面に表示したり、本体が実行した計算の結果、絵、図などを表示する出力装置のひとつだ。

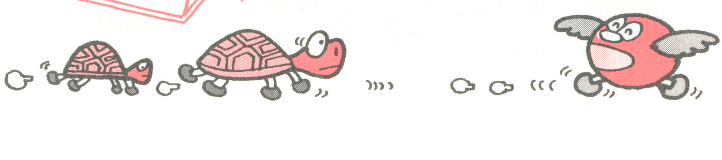


本格的なパソコンになると、もっといろいろな周辺機器があるけど、これからファミコンをマスターしようというきみに、とりあえず必要なのは、作ったプログラムをテープに保存しておくためのカセットレコーダくらいかな。ゲームで遊ぶときは、ジョイスティックもあると便利かな。

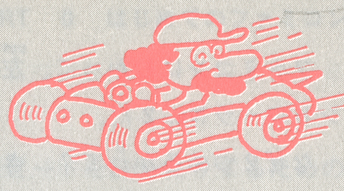


■周辺機器のいろいろ





No.3 ゲームするためにキャラクタを動かす

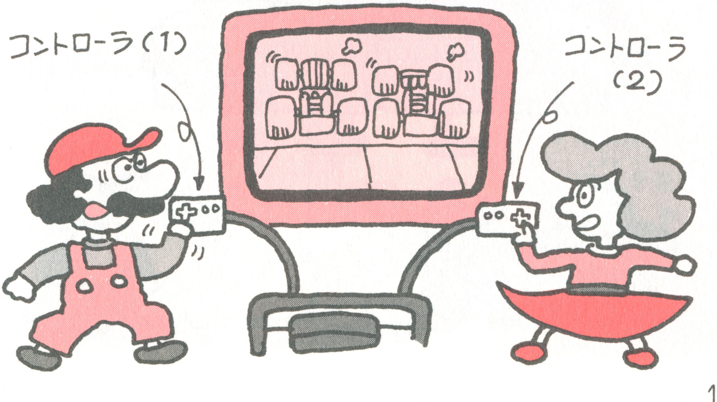


まず手はじめはコントローラ やがてジョイスティックがほしくなるぞ

♥2つのコントローラは敵(てき)と味方(みかた)だ

さあ、きみたちにとってはファミコンゲームのいちばんたいせつな装置(そうち)といっていいのが、コントローラだ。なぜなら、コントローラがなければ、ゲームのキャラクタを動(うご)かすことができないからだ。

コントローラは、ファミコンに指令(しれい)を出(だ)して、いろいろなキャラクタを動(うご)かす装置(そうち)で、ファミコンには2つついている。



ファミコンにむかって、左側がコントローラNo. 1、右側がコントローラNo. 2だ。

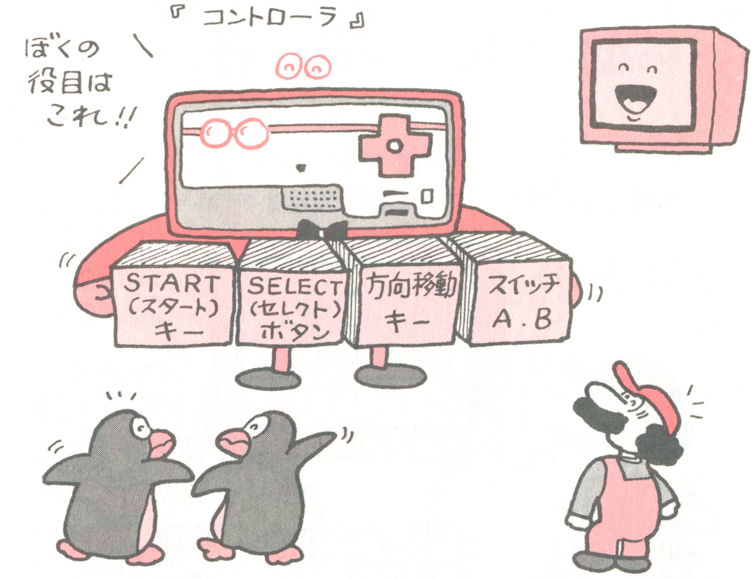
なぜ2つあるかって？　きまってるじゃないか。2人で遊べるようにさ。パソコンゲームの遊び方には、キーボードのキーをたたいてキャラクタを動かしたり、命令を出して遊ぶ方法もあるけど、これでは2人では遊べない。敵、味方に分かれて戦争するなんてゲームには、2つの装置があったほうが、ぜったいにいいものね。

♥ゲームによってちがうスイッチA、B

コントローラについているボタンの役わりを説明しておこう。

・START(スタート）キー

コントローラNo. 1についている。このボタンを押すと、ゲー



ムがスタートする。

・SELECT（セレクト）ボタン

画面を変化させたり、ゲームのスピードを決めたりするためのボタン。1人で遊ぶときと2人で遊ぶときとでは、このセレクトボタンの役わりがちがう。また、ゲームによってもちがうので、説明書をよく読もう。

・方向移動キー

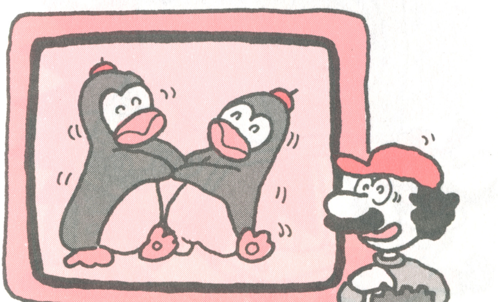
左側の上下左右に矢印の指示のあるキーが、なにをかくそうキャラクタを移動させるためのキーだ。きみたちは、このキーをバンバン押して、ゲームに熱中するというわけだ。

矢印はキャラクタの動く方向だ。

・スイッチA、B

このスイッチの使い方は、よく説明書を読むしかない。ゲームによって、それぞれ使い方がちがうからだ。ひとつ例をあげておこう。ロードランナーというゲームでは、Aを押すと右側に穴を掘り、Bを押すと左側に穴を掘る。

方向移動キーでキャラクタを動かすだけでなく、画面にもどんどん入りこんで、お



●ポイント

コントローラはたいせつにあつかおう。ゲームに熱中するあまり、コントローラをこわしたりすると、とりかえるのはたいへんだぞ。

もしろさをグーンとアップさせるというわけだ。

• **MIC、VOLUME**

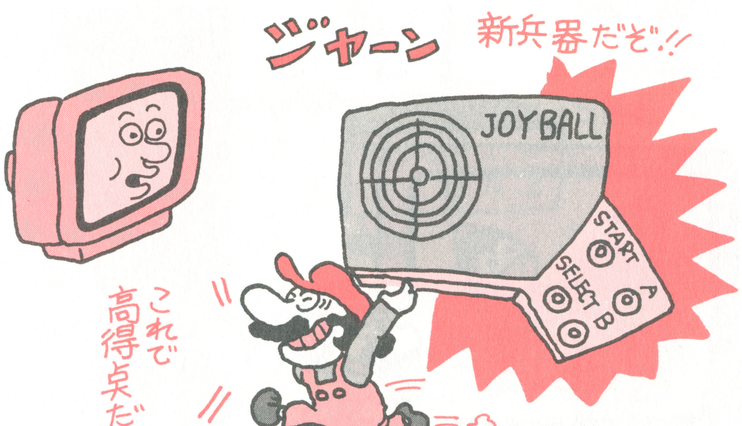
コントローラNo. 2には、STARTキーとSELECTボタンはついていない。そのかわりにMICとVOLUMEという機能がついている。これは、自分の声を使って遊ぶゲームに使うんだ。たとえば、ミュージックボードとMICを接続すると、カラオケもできるんだぞ。

♥速いゲームにはジョイスティックやジョイボール

2つのコントローラは、ファミコンを買ったときについてくるから、ほとんどの人がはじめは、このコントローラを使って遊ぶだろう。

でも、だんだん腕まえもあがってくると、きっとほしくなるのが、ジョイスティックやジョイボールだ。

腕まえがあがってくると、当然、ゲームのスピードも速いほうがおもしろくなる。そうすると、たとえばスターフォースのよう



な連射打ちなんていうのがあるゲームだと、コントローラのボタンをすごい速さで押さなくてはならない。とてもくたびれてしまうんだ。

♥なんと1秒間に15回も自動的にボタンを押す

その点、ジョイボールは圧巻だ。だいたい、きみは1秒間になん回、コントローラのボタンを押せるだろうか？　親指のつめでこするとか、定規でピシャピシャたたくとか、それなりのくふうをしている人もいるらしいけど、ジョイボールにはかなわない。なんと、1秒間に15回も、自動的にボタンを押してくれるのだから。この自動連射装置の威力の前には敵なしといったかんじだ。

4方向と8方向の切りかえ装置つきなんていう、便利モノジョイスティックもある。

ただし、注意しなければならないことがある。これらのジョイスティックやジョイボールは、ゲームによって使えるものと使えないものがある。買うときには、どのゲームに使えるのかをよく聞いてから、買わないといけない。

●ポイント

ゲームのスピードをあげるには、ジョイスティックやジョイボールがほしくなるが、これらを買うときにはどのゲームに使えるかをよくたしかめてから買おう。



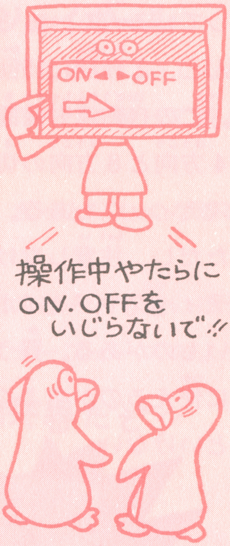
よくおぼえておこう！

♥バックアップスイッチの使い方

電源スイッチを切っても プログラムを記憶しているよ

バックアップスイッチは、とっても便利なスイッチだけど、操作中にやたらにONにしたり、OFFにしたりすると、へんな動作になったりすることがあるから、使い方には注意しよう！

ファミリーベーシックのカセットは、バックアップスイッチをONにしておけば、ファミリーコンピュータの電源を切っても、ベーシックのプログラムを記憶させておくことができるよ。



スタート画面のときは、画面でON、OFFを教えてくれるけど、ベーシックモードのときは、次のようにしよう。

1．SYSTEM↵ として、ゲームベーシックのモードにもどす。

2．数字キー3でENDを選べば、スタート画面にもどる。

3．ファミリーベーシックのカセットのバックアップスイッチをONにして、ファミコンのスイッチを切る。



PART 1

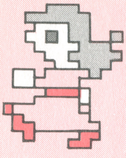
はじめてのファミリーベーシック

ファミコン・マシンは、ゲームを遊んでいるだけでも楽しいが、ただゲームをするだけではなく、ほかの仕事もさせてみれば、もっともっと楽しみは大きくなるよ。

ファミコンをゲーム機だけに使うなんて、なんとなくものたりなく感じないかい?

まずは、ファミリーベーシックをスタートさせよう!







1.1 ファミリーベーシックの接続

ファミリーベーシックをマスターすれば自分だけのゲームも自由自在だ

♠計算させたり絵を描かせたり音楽を演奏させたり

とにかくファミコンゲームはおもしろい。おもしろくておもしろくて、一度やったらやめられない。たしかに、そうなんだ。

だけど、いくらゲームの腕まえがあがっても、それだけでファミコンのすべてがわかったと思っていたら、おおまちがいだぞ。



ファミコンには、実は、まだいろいろな機能があるんだよ。知っていたかな？ そのいちばん大きな機能が、ファミリーベーシックというやつなんだ。

パソコンゲームマニアのきみなら、一度くらいは、ベーシック（BASIC）ということばを聞いたことがあるだろう。

コンピュータは、機械だけあったって、かってに動いてくれるわけじゃない。ゲームにしても、計算にしても、また絵を描いたり、音楽を演奏したりするのも、人間が、こうしろ、ああしろと命令を与えなければなにもしないよ。

いったん命令を与えてしまうと、今度は、人間よりも速く確実に、すばらしい仕事をしてくれる。つまり、ぼくたち人間がわとしては、その命令の与え方をマスターしなければならないというわけだ。

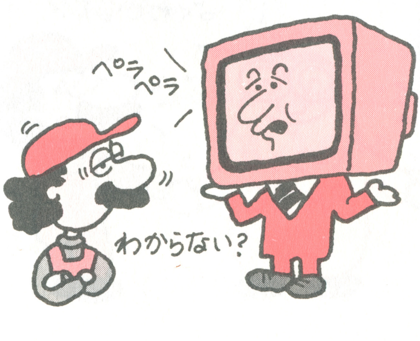
♠ファミコンと話し合うことばがファミリーベーシックだ

では、どうやって命令を与えるのか。人と人とが理解し合うために、ことばがある。人とコンピュータが意思を通じ合わせるにも、ことばが必要だ。それは、ぼくたちがふだんしゃべっている

●ポイント

ベーシックをおぼえてしまえば、ファミコンだけに使えるのではない。

ほかのパソコンにも応用して使えるから、いまのうちにマスターしておこう。



ようなことばじゃない。コンピュータが理解できるコンピュータ言語ということばで、意思を通い合わせるんだ。

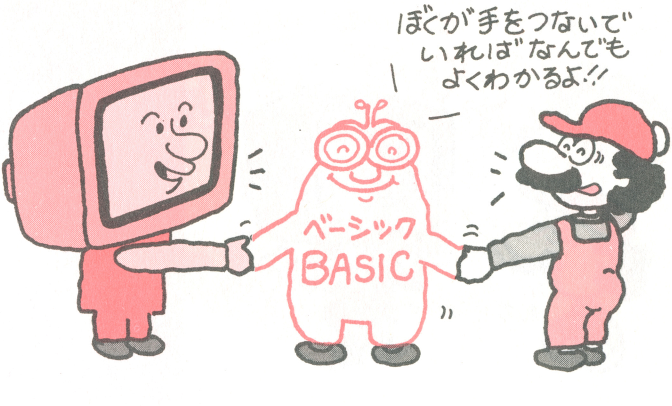
ベーシックというのは、そのコンピュータ言語のひとつで、しかも、いちばん人間のことばに近く、やさしいことばなんだ。

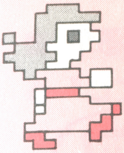
初心者のきみでも、真剣に取り組めば、容易にマスターできることばだよ。いったんマスターしてしまえば、それらベーシックのことばをどんどん使いこなして、自由自在にプログラムを作れるようになる。

ゲームだってなんだって、自分のオリジナルのものが作れるわけだ。

カートリッジのゲームで遊ぶだけでなく、ベーシックにも挑戦しようという、好奇心旺盛なきみのことだ。いっしょうけんめい勉強すれば、あっという間におぼえてしまうと思うよ。そして、将来、もっと高級機種のパソコンを扱うときや、また、機械語みたいなむずかしいコンピュータ言語を扱うときにも、きっときっと役に立つはずだ。

さあ、がんばろう。



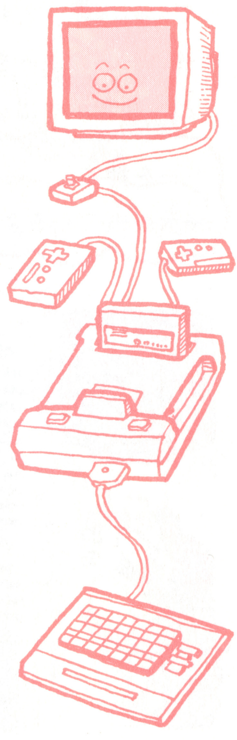




1.2 ベーシックをスタート

**テレビ画面に絵が出てきたら
ベーシックが動作したという合図だよ**

♠キーボードから出ている線を本体につないで



さて、いよいよファミリーベーシックへの第一歩をふみだすぞ。用意はいいかな？

まず、本体の電源は切っておくんだったね。キーボードから一本の線が出ているね。これがエキスパンドコネクタ（機能拡張コネクタ）とよばれる線で、この線をファミコンの本体に差し込む。本体の差し込み口はうしろにある。ピンが見えるからすぐにわかると思うよ。

次に、かんじんのベーシック用カートリッジを本体に差し込む。このカートリッジこそ、ファミリーベーシックの頭脳だ。キーボードから打ち込まれた命令やことばは、この頭脳がすべてキャッチし、処理してくれるんだ。

さあ、すべて準備はととのった。スイッチを入れる順番は、テレビからだったね。

それ行け、スイッチON!!

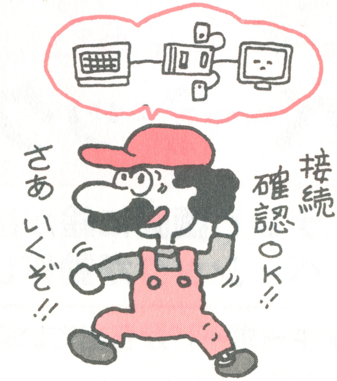
♠スイッチONして絵が出たら、それからどうする？

スイッチON!!

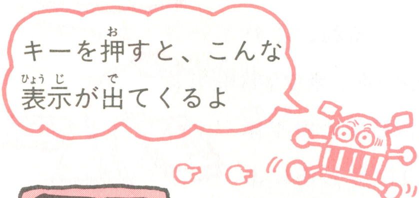
出てきた、出てきた。テレビの画面にコンピュータの絵が出てきたね。これは、ベーシックが動作しはじめたという合図だ。でも、これではまだ、プログラムを入力する準備はできていない。いくつかの手つづきが必要だ。

では、次に、どれでもいいから、キーボードのキーを押してみよう。

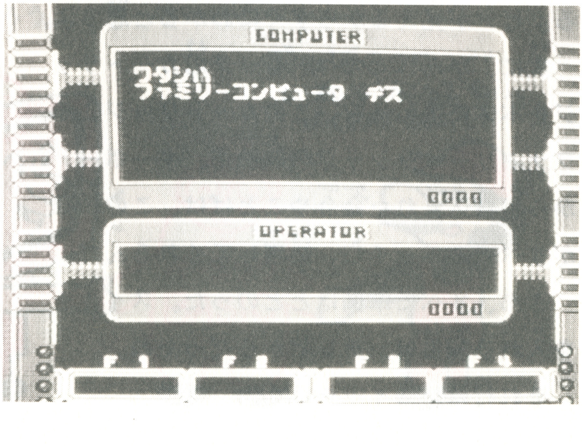
どんな表示が出たかな？











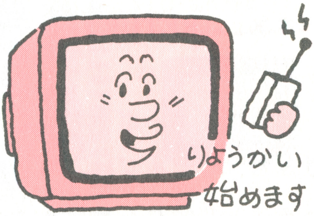
■RETURN（リターン）キーの役わり

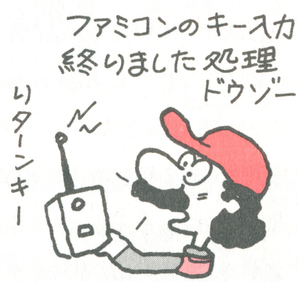
♠"コンピュータさん、あなたの番だよ"の合図

アナタ、つまりきみの名まえをキーボードから入力する。このとき、名まえのぜんぶを入力してもかまわないけど、ふつうは1文字でいい。じぶんの名まえの頭文字を入力したらいいよ。太郎ならT、健太ならKというふうにだ。

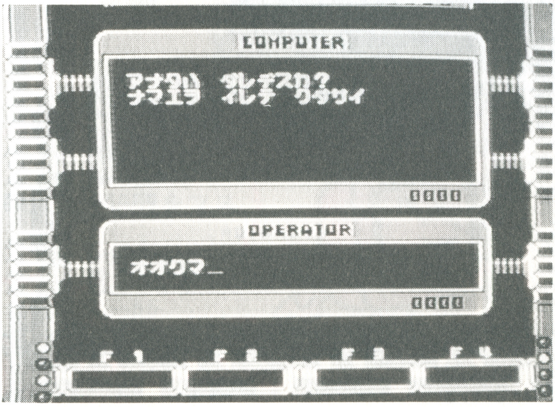
かんじんなのは、その次。名まえの頭文字をキー入力したあと、かならず RETURN（リターン）キーを押すこと。

この RETURN キーは、これからもたくさんでてくるけれど、だいじなキーだよ。





RETURN キーを押すのをわすれないでね





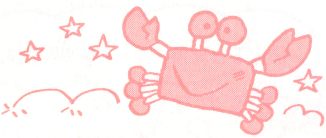
その役わりは、「コンピュータさん、こちらの作業は終わりましたよ、どうぞ処理してください」という合図なのだ。

さあ、RETURN キーを押してごらん。はたして、「リョウカイ！」「アナタハ　Tサンデスネ」と表示されたはずだ。

どうだ、キーボードから入れたきみのメッセージは、ちゃんとファミコンに伝わっただろう？

「GAME BASICニシマスカ？」これが次に出てくるメッセージだ。

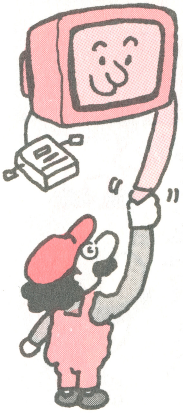
このGAME BASICこそ、これからいっしょにマスターしていこうというテーマで、ゲームのプログラムを作成するベーシックだ。

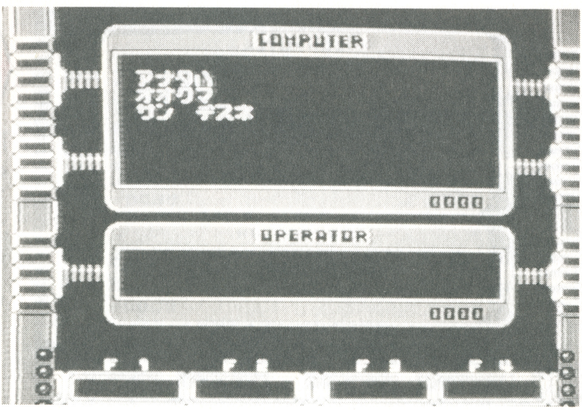


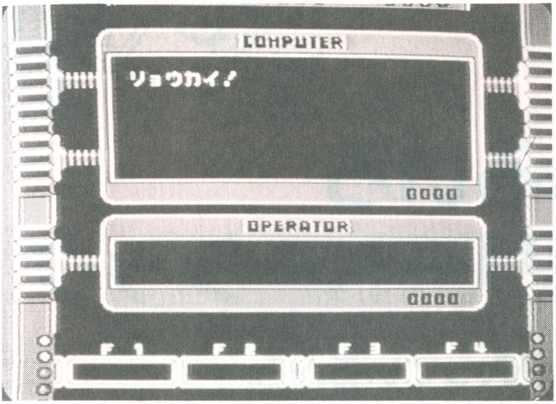
●ポイント

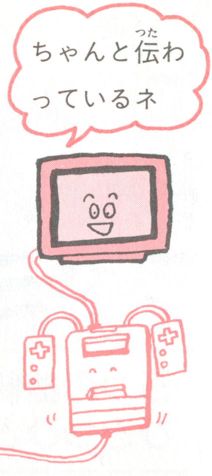
コンピュータには、入力、出力ということばがよく出てくる。

入力は、キーボードのキーをたたいて、コンピュータにことばや合図を入れること、出力は、入力で入れたものの結果を画面に呼び出して表示させることと考えておこう。

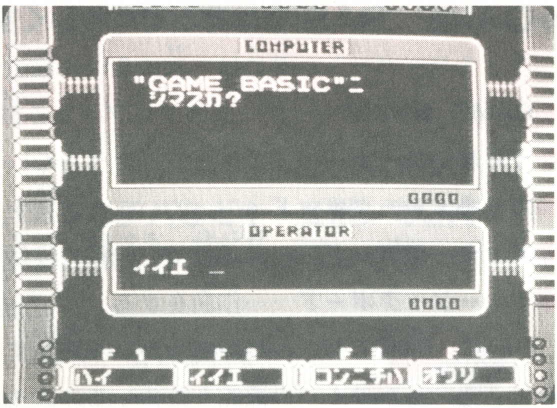


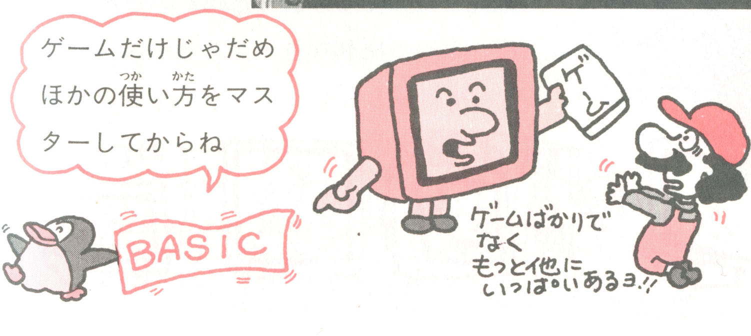


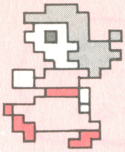

COMPUTER
リョウカイ!
0000
OPERATOR
0000
F1 F2 F3 F4


ちゃんと伝わっているネ


わーい。ゲームをしようよ
ゲーム


COMPUTER
"GAME BASIC"ニ
シマスカ?
0000
OPERATOR
イイエ _
0000
F1 F2 F3 F4
ハイ
イイエ
コンニチハ
オワリ


ゲームだけじゃだめ
ほかの使い方をマスターしてからね
BASIC
ゲーム
ゲームばかりでなく
もっと他に
いっぱいあるヨ!!





1.3 機能いろいろファミリーベーシック

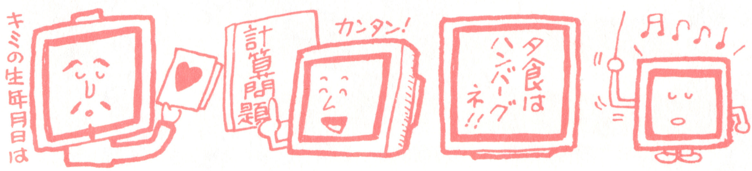
5つの機能が入っているから ひとつひとつたしかめてみよう

♠GAME　BASICはあとでくわしくするとして

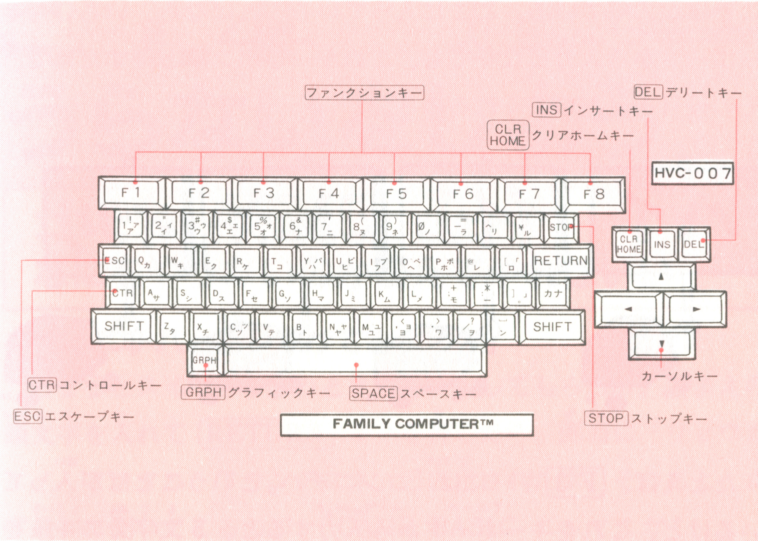
ファミリーベーシックには、次の5つの機能が入っている。

- **GAME　BASIC**……ゲームのプログラムを作成する
- **うらないコーナー**……コンピュータうらないをする
- **カリキュレータボード**……いろいろな計算をする
- **メッセージボード**……伝言板や記録板の役目をする
- **ミュージックボード**……作曲をする

GAME　BASICは、これからくわしく勉強するとして、手はじめに、そのほかの4つの機能をためしてみようと思うのだが、その前に、ファンクションキーという便利なキーがあるから、それを頭に入れておこう。



■ファミコン・キーボード



■ファンクションキー

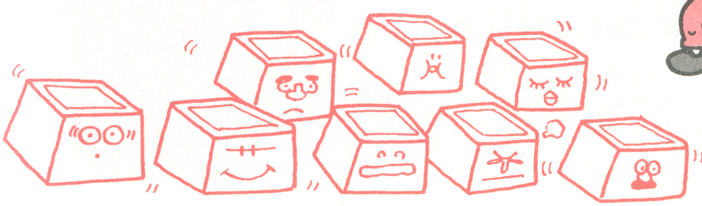
♠ハイ、イイエ、コンニチハなどキーをポンでOK

キーボードのいちばん上に、F 1 から F 8 まで並んだ、8つの赤いキーがあるね。

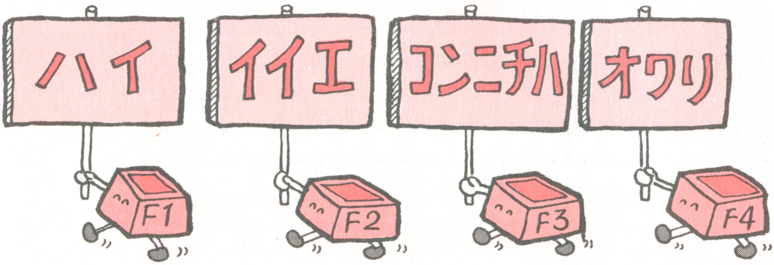
これはファンクションキーとよばれるキーで、とても便利なキーなのだ。

画面の下の方に、なにか文字が出ていないかい。「ハイ」とか「イイエ」などだ。

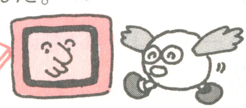




そう、このファンクションキーには、あらかじめよく使われることばが入っている。ここに入っていることばなら、いちいちキーボードから入力しなくても、このキーをポンッと押すだけで、その役目をはたしてくれるのさ。



たとえば、[F 3]キーには「コンニチハ」ということばが入っているけど、これをキーボードから入力すると、5文字分のキーを押さなければならない。でも、[F 3]キーをポンと1回押すだけで、この5文字をファミコンに伝えてくれるんだ。





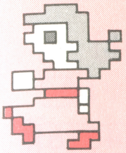
また、このファンクションキーにあらかじめ設定されていることばを変更して、じぶんがよく使うことばに設定しなおすこともできる。

ただし、この変更ができるのは、ベーシックでプログラムを作成するときだけ。つまり、GAME BASICを選んだときだけだ。

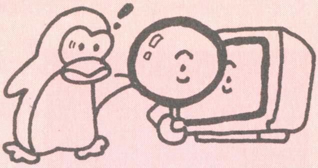
くわしくは、本格的にベーシックの勉強に入ってから説明しよう。

●ポイント

ファンクションキーには、じぶんがよく使うことばを入れておくことができるから、利用すると便利だ。



1.4 うらないをしよう



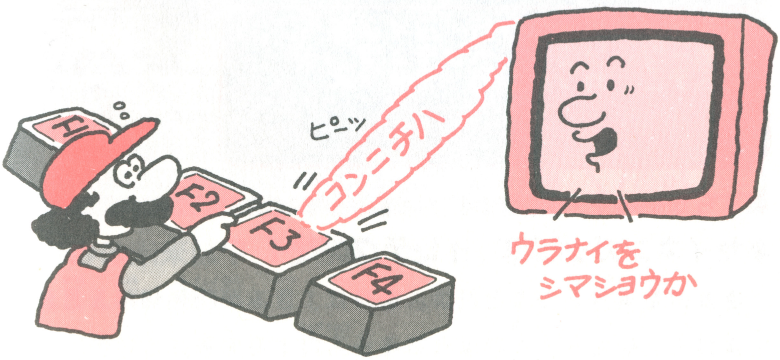
コンピュータが聞いてきたら なにをやりたいのか答えてやればいい

♠コンニチハと話しかけてみようか

ちょっとスイッチONのときを思い出してもらおう。名まえの頭文字を入れて RETURN キーを押したら、画面に「リョウカイ！　アナタハ　Tサンデスネ」と表示されたネ。そのあと、まず最初に聞いてくるのが、「GAME BASIC ニシマスカ？」

ここで、ベーシックをするなら、F 1 キーを押す。つまり、「ハイ」と答えるわけだ。

ほかの機能を使いたいときは、「イイエ」の F 2 キーを押せばいい。

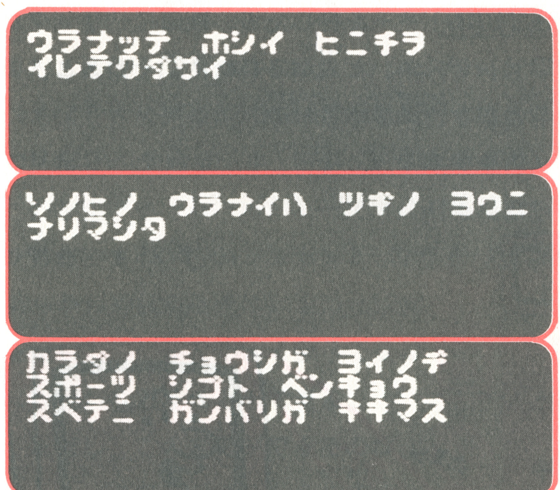


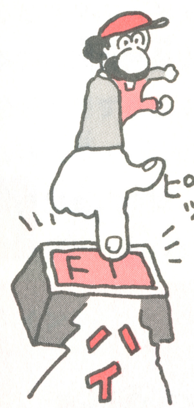
「イイエ」と答えると、入っているメニューの順番に、
「カリキュレータボードニ　シマスカ？」

そのたびに、「ハイ」（F１キー）か「イイエ」（F２キー）で答えていけばいいよ。

そして、どういうわけだか、はじめでも途中でも、コンニチハキー（F３）を押して画面に話しかけると、
「ワタシハ　ウラナイガ　デキマス。Tサン　ウラナイヲ　シマスカ？」と、きみに話しかけてくるからおどろきだ。

うらないをしてほしかったら、F１キーで「ハイ」と答える。RETURNキーを押さなければ、答えたことにはならないぞ。





■うらないのキー入力方法

♠セイネンガッピヲ　イレテクダサイ

きみの生年月日のことだよ。かりに、きみが昭和46年５月８日生まれだとしたら、次のようにキー入力するのだ。

[S][4][6][/][5][/][8][↵]

[↵]はリターンキーのこと。くれぐれも(RETURN)キーを忘れないように。

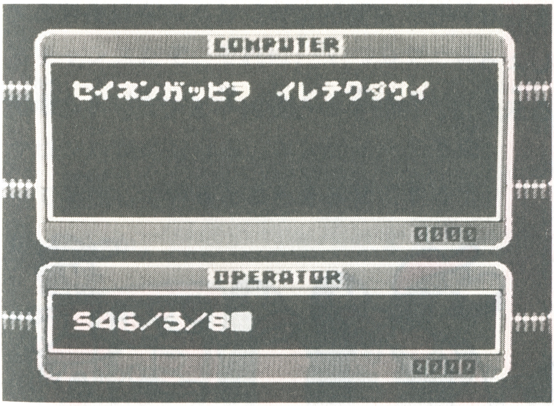
はじめのSは昭和の略だね。もし、おじいちゃんやおばあちゃんのうらないをしてあげたいときは、大正生まれならT、明治生まれならMを、いちばんはじめにキー入力だ。

●ポイント
そのひとの生年月日とうらないたい年月日を入力する。
[/]（スラッシュ）を忘れるな。

また、西暦でもできるぞ。昭和46年は1971年だ。
次のようにキー入力すればいい。

[1][9][7][1][/][5][/][8][↵]

[/]（スラッシュ）は、年月日のくぎりを示す記号だから、入れ忘れると、なんのことだかぜんぜんわからなくなってしまうぞ。





♠きみの未来は？　うらないたい日を入力

たとえば、S65年1月1日のきみには、いいことが待っているのか、それともあまりいいこともなさそうか？　まあ、とにかく

キー入力してみよう。

S 6 5 / 1 / 1 ↵

♠いいことがなくてもくよくよするな

パッパカパーン、パ、パ、パ、パッパカパーン。ファンファーレの音とともに、うらないの結果が、画面に表示される。

信じるも信じないも、きみの自由だが、いいことがないと出ても、くよくよするなよな。

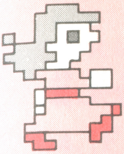
もっとうらないを続けるかどうか、ファミコンが聞いてくるから、続けるなら「ハイ」、終わりにするなら「イイエ」だ。

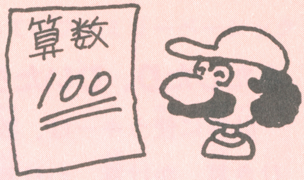
また、別の人をうらなってみたいときは、「イイエ」を押すと、はじめのスイッチONの状態の画面にもどるから、同じ手順でもう一度うらないコーナーを選べばいい。

いつのうらないをしようかなあ、なんて、あまり長い時間迷っていて、キー入力をしないと、自動的にコンピュータは終了してしまうから、注意しよう。

また、うらなえる未来は、1968年から2924年までだ。すごいだろう。







1.5 カリキュレータボード 算数の勉強する？

ファミコンはゲームだけじゃない
勉強もするんだというところをみせよう

♠えんぴつの代わりには羽根ペンとインクつぼ

画面で算数の勉強をするなら、カリキュレータ（計算）ボードを使えばいいのだ。

計算がにがてというひとには、勉強するチャンスだぞ。それに、「いつもゲームで遊んでばかりいて」なんてうるさいママゴン対策にもなる。コンピュータを使って勉強しているところをみせてやろうじゃないか。

●ポイント

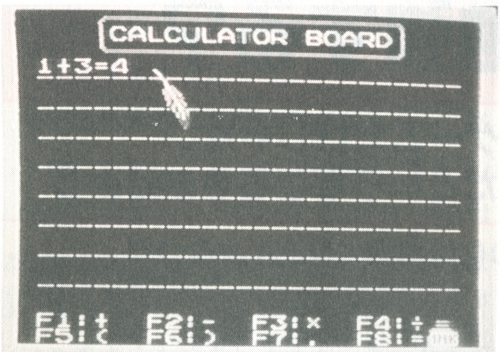
カリキュレータボード、メッセージボード、ミュージックボードをまとめて「各種ボード」ということもある。そして、各種ボードの主役は楽しい羽根ペンとインクつぼだ。

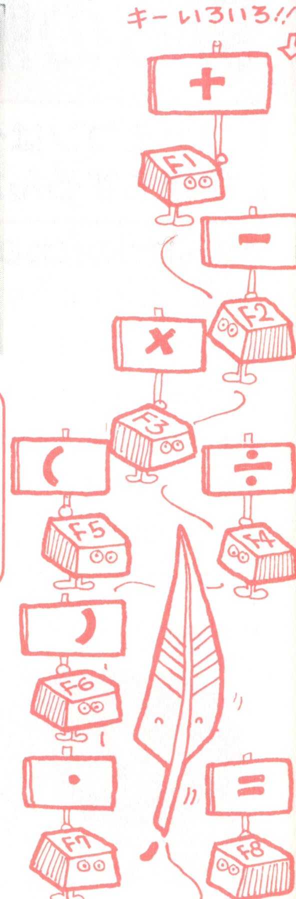


♠たしたり、ひいたり、かけたり、わったり

計算に使うキーは、数字キーとファンクションキー。たし算、ひき算、かけ算、わり算の４種類の計算ができるよ。

ファンクションキーのひとつひとつが、計算記号の役わりをするんだ。画面の下に表示されているね。





F１……＋	F５……(（左カッコ）
F２……－	F６……)（右カッコ）
F３……×	F７…….（小数点）
F４……÷	F８……＝（イコール）

これ以外のキーを押すと、　エラー（デキマセン）といわれてしまうぞ。

さて、羽根ペンは画面の左上にきているね。ひとつ計算してみよう。次のようにキー入力してみよう。

１.数字キーを押す。たとえば１。

２.ファンクションキーを押す。たと

えば、[F 1]。

3.数字キーを押す。たとえば3。

4.[F 8]キーを押して、答えを求める。

こうやってキーボードからキー入力すると、羽根ペンが自動的に移動するから、どんどん計算していけばいいよ。

♠[RETURN]キーは改行の役め

前に、[RETURN]キーは、「入力終わり、処理どうぞ」を意味するキーだと説明したね。

このカリキュレータボードには、自動的に計算する機能が入っているから、計算処理は、キー入力さえすればかってにファミコンがしてくれる。

ここでの[RETURN]キーの役わりは、改行、つまり、行を変えてくれるんだ。

ためしに、今計算したそのままの画面で、[RETURN]キーを押してごらん。

羽根ペンが、一段下がった行の先頭にきただろう。



キチンと並んで
列を変えるから

♠じぶんで計算問題を作ってパワーアップ

計算はにがてというきみも、ファミコンを使って勉強すれば、グーンと実力アップだぞ。

画面に式だけを作っておいて、その答えを紙に書いておく。あとから＝（イコール、[F 8]キー）を書き込んで答えを確認しよう。

50×2

38×46

$95 - 72$

$6.5 + 9.3$

$54 \div 6$

$2 \times (8 + 5) - 16$



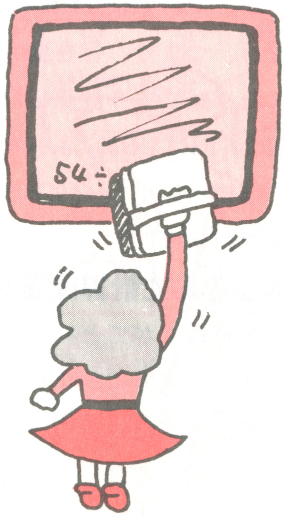
あとからイコールを書き込むのは、矢印キーを使う。矢印のある方向に羽根ペンが移動するから、計算問題のうしろに羽根ペンを移動させてから、［F 8］キーを押せば、答えが表示されるよ。

■カリキュレータボード操作のあれこれ

♠文字を消したり付け加えたりその他いろいろ

●計算の範囲

計算できる数の範囲は、99,999,999から−999,999まで。この範囲を越えると、オーバーフローエラーになって、「計算デキマセン」ということだ。



●このボードをやめるなら

カリキュレータボードを終了したいときは、左上にある［ESC］（エスケープ）キーというのを押せばいい。もとの機能を選択するところへもどる。

●文字を消したいなら

矢印キーの上の［DEL］（デリート）キーを押してごらん。1回押せば、1文字消えるよ。

●文字を追加したいなら

[INS]（インサート）キーを押すと、1文字分あくから、そこに文字を追加しよう。

●画面左上のスタート位置にもどりたいなら

[CLR／HOME]（クリアホーム）キーを押す。

●画面全体を消したいなら

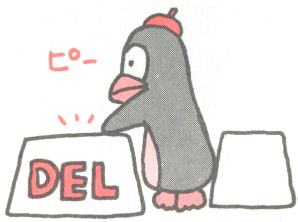
[SHIFT]（シフト）キーと[CLR／HOME]キーを同時に押す。左上のスタート位置にもどるよ。もう一度はじめからやり直せる。

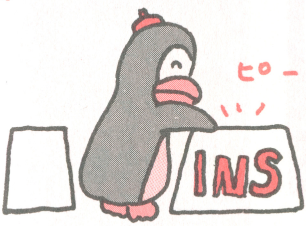
●カリキュレータボードに書き込んだ数字や文字を消したいなら

[RESET]（リセット）ボタンを押すか、電源を切る。画面に書かれた文字のことではないよ。計算するために本体に書き込まれたデータのことだよ。

5 2 6 × 5 8 9 ＝ 3 0 9 8 1 4
5 2 6 × 5 9 _ 3 0 9 8 1 4
5 2 6 × 5 9 ＝ 3 1 0 3 4

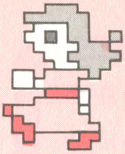
5 8 9 を59にしたい。まず羽根ペンを移動して [DEL] キーを押す。＝も消えるからもう一度＝を入れると答えが出るヨ

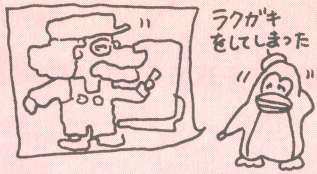




5 2 6 × 5 8 9 ＝ 3 0 9 8 1 4
5 2 6 × 5 8 □ 9 _ 3 0 9 8 1 4
5 2 6 × 5 8 1 9 ＝ 3 0 6 0 7 9 4

5 8 9 を 5 8 1 9 にしたい。羽根ペンを 9 に移動し、[INS] キーを押して 1 を入れる。もう一度＝を入れよう





1.6 らく書きしよう 絵も描こう

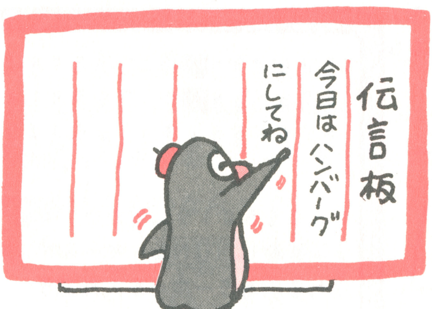
メッセージボードは自由な伝言板
文字や記号も書いたり消したり

♠数字キーでグラフィックキャラクタも出せる

このボードでは、らく書きをしたり絵を描いたりすることができる。

表示できる文字数は、横に24文字、たてに17文字の合計408文字まで。キーボード上に書かれている文字や記号、数字はすべて自由に使える。

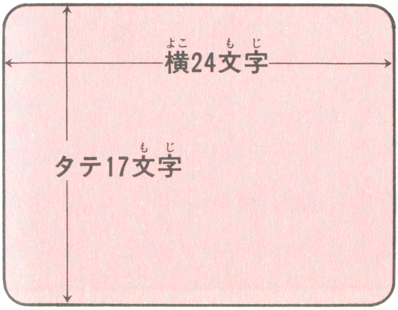
さらに、グラフィックキャラクタも13個用意されているから、文字や記号と組み合わせてみるのも楽しいよ。



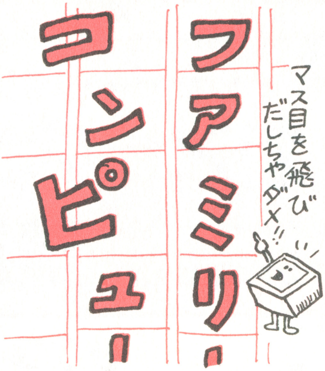
●ポイント

各数字キーがどのグラフィックキャラクタに対応しているかを、まずおぼえてしまおう。

■メッセージボードに表示の文示数









このグラフィックを表示するには、まず(カナ)キーを押して、カナモードにする。羽根ペンの色は、赤から緑色に変わるよ。それから、(GRPH)(グラフィック)キーを押しながら数字キーを押すと、次のように対応して出てくるからおぼえておこう。

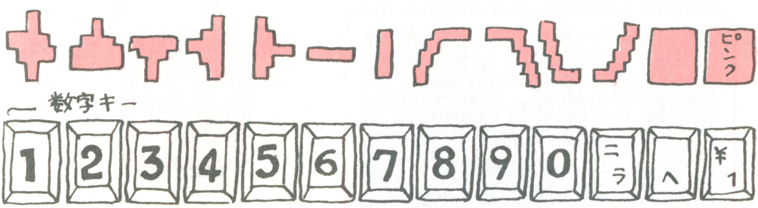


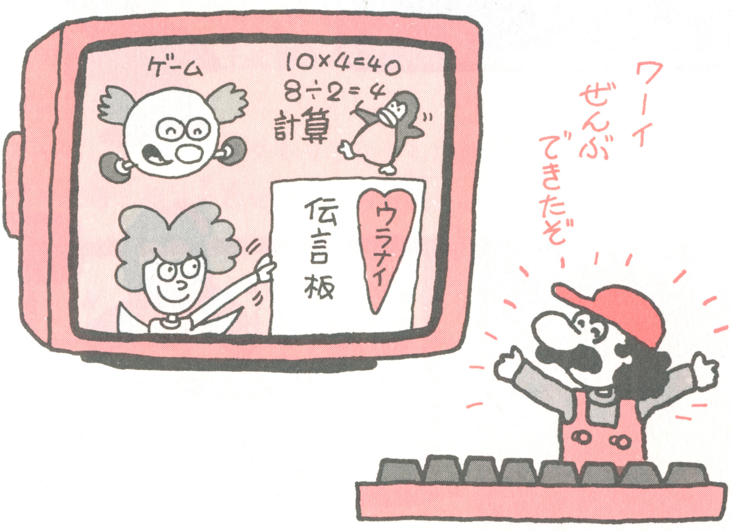
■数字キーとグラフィックキャラクタの対応

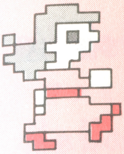
このグラフィックキャラクタを使って、わくを描いたり、線を描いたり、組み合わせて模様を作ってみるのも楽しい。

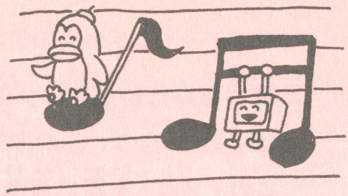
また、「チョット　ヤマシタ　クンチヘ　イッテクルヨ」とか、「コンヤノ　オカズハ　ハンバーグニ　シテネ」なんて、おかあさんにメッセージを書いておくのにも便利だ。

文字の追加や消し方は、カリキュレータボードのときと同じだよ。









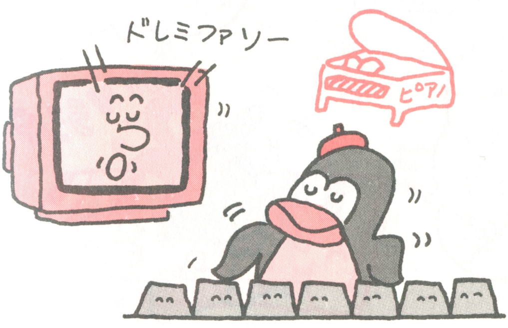
1.7 ミュージックボードはファミコン演奏

きれいなメロディを楽しむのも
へたな作曲してみるのもきみしだい

♠3つの音域を使えば三重奏だってできる

　ミュージックボードを選択すると、画面にボードが表示されるね。この画面は、4行にわかれている。上の行から順に、羽根ペンが右に移動しながら、「ドレミ………」というぐあいに、音符をカタカナで書き込んでいくんだ。

　さらに、この1行は、それぞれ3段にわかれている。これで三重奏ができるわけなのだが、上から高音、中音、低音の3つの音域にわかれている。はじめは上段に羽根ペンがあるから、二重奏、



●ポイント

　はじめはでたらめでいいから、ドレミファ……を適当に入れて、音を鳴らしてみてもいいではないか。

三重奏をしたいときは、矢印キーで羽根ペンを移動させて書き込めばいい。

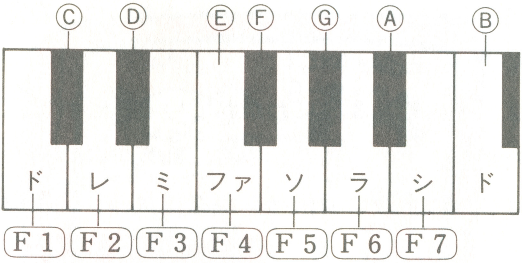
　このマスは、実際に演奏をはじめると移動して、今どこの音を演奏しているのかがわかるしくみになっているんだ。

　では、さっそく音を入力してみることにしよう。

♠ファンクションキーが音符がわり

　ファンクションキーの［F1］～［F7］までが、音符のド～シの役をする。［F1］キーを押すと画面にはカタカナでドが、［F2］キーならカタカナでレが表示されるというぐあいだ。

　半音上げたいときには、［SHIFT］キーを押してファンクションキーを押す。半音上がった音は、アルファベットのC～Bで表示される。ドの半音上がった音はC、レならDだ。下の図をみておぼえておこう。



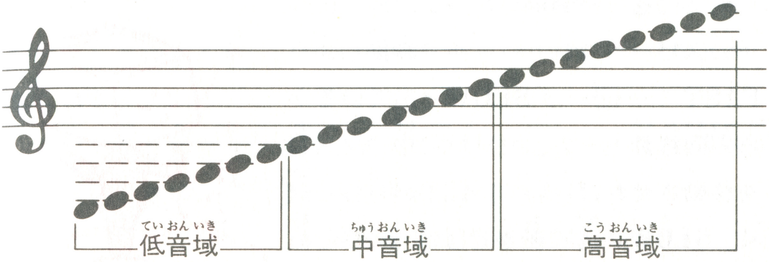
Cは、［SHIFT］キー＋［F1］キー。Dは、［SHIFT］キー＋［F2］キー。ほかの半音もためしてみよう

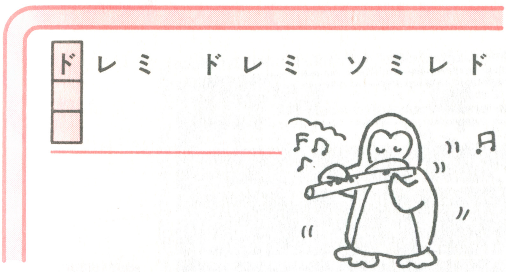
Eはミの半音上がった音、つまりファと同じ、Bはシの半音上がった音、つまりドと同じだ。音楽の時間にならったよね



出せる音の範囲

ファミコンは、低音域、中音域、高音域の三つの音域が出せるから、組み合わせて演奏してみよう。



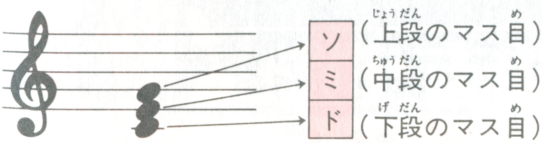


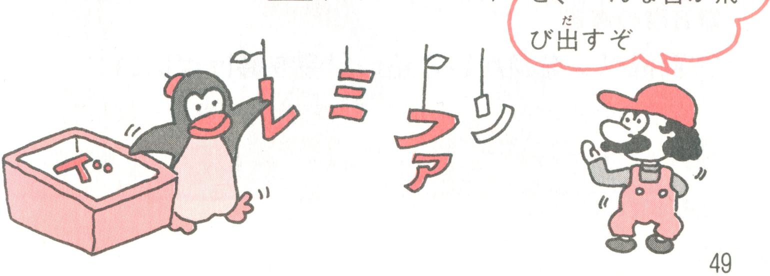
●ポイント

作曲中は、マス目は移動しないよ。音を伸ばしたいときには、その分だけスペースをあければいい。

たとえばドミソの三重和音

3つのマスをぜんぶうめれば三重和音だ。



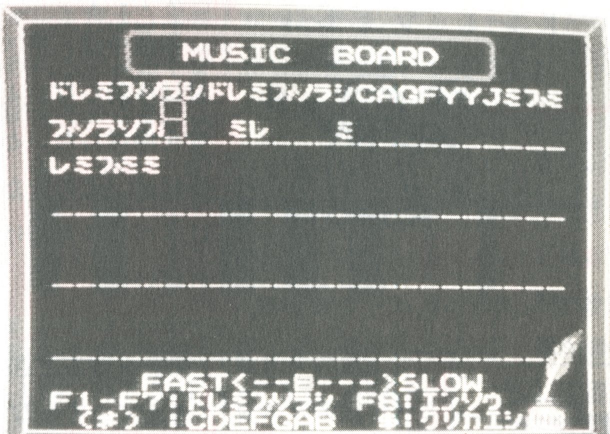


♠F8キーでミュージック演奏スタート

（F 8）キーは演奏をスタートさせるキーだ。四角いマスが移動して、どこを演奏中かを教えてくれているね。

演奏の速さを調節したいときは、画面下のスピードコントロールで調節しよう。FAST〈……□……〉SLOWと出ているから方向移動キーで、カッコの中のマス目を移動させればいい。FASTの方なら速く、SLOWの方に動かせばゆっくりと演奏してくれる。







■音階の修正

♠羽根ペンを動かして正しい音を書けばいい

音階の修正はかんたんだ。方向移動キーを押せば、羽根ペンが動いてくれるから、訂正したい箇所へ移動させて、書いたり消したりすればいい。

演奏中に修正したいときは、CLR／HOME キーを押し、羽根ペンを移動させて修正する。

また、演奏中に SPACE キーを押すと、押したところで四角いマスが止まってくれる。方向移動キーの左（◀）を押すと左に、右（▶）を押すと右に、マス目が動いて、その箇所だけを演奏してくれる。

これで音を聞きながら修正すると便利だよ。ふたたび演奏を開始したいときは、F 8 キーを押せばいいんだ。

画面をすべて消すのは、SHIFT キーと CLR／HOME キー。

羽根ペンをスタート位置にもどすのは、CLR／HOME キー。

INS キー、DEL キーは、挿入、消去。終了は ESC キー。他のボードのときと同じだね。

$マークを入力するとくり返し演奏されるよ





このミュージックボードと、コントロールキーⅡのMICをつなげば、カラオケもできるよ

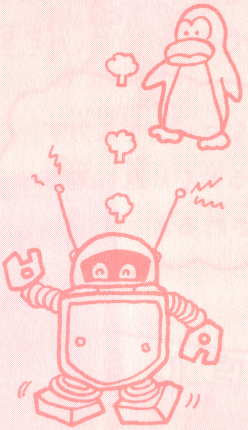
きみ知ってるかい？

♠ゲームだけじゃないんだよ

ファミコンに信号を送って ロボットを自由に動かそう

ファミコンで、てっていてきにゲーム遊びするのは、もちろん楽しいけれど、ゲームだけではなく、ほかにもいろいろつなげて、もっともっと楽しむことができるんだぞ。

光線銃やロボットをつないで、遊びまくるのも楽しいよ。



ファミコンからの信号で、自由に動かせるロボットなんてどうだい？　ファミリーコンピュータロボットっていうんだ。

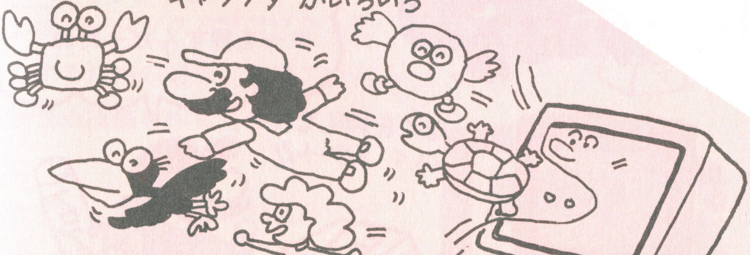
ファミコンロボットの目はすごいよ。光センサーが組み込まれていて、テレビの画面から出る光をキャッチするから、キミがファミコンに信号を送れば、ロボットは思いのままに動くというわけだ。こんな遊び方は、これからもどんどんふえていくだろうよ。

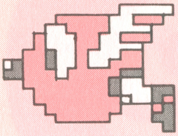


PART 2 ゲームベーシックの初歩

ゲームベーシック(GAME BASIC)は、いままでやってきた各種ボードとは、ちょっとちがうぞ。

ゲームベーシックということばを使って、ファミコンに命令すれば、いろいろなキャラクタが表示されたり、キャラクタが動いたりして、ゲームが始められるぞ！







2.1 ゲームベーシックする？

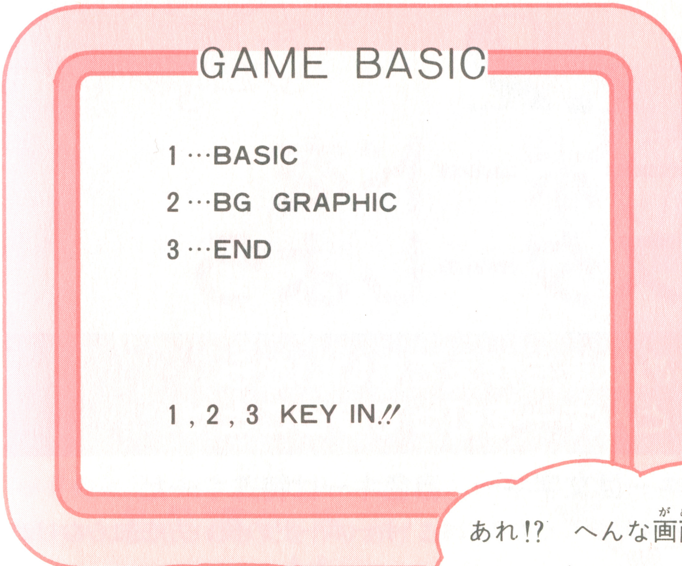
画面で"BASIC"を選択したら さあ、いよいよゲームへの入口だ

♣数字キー1はベーシック入力のキーワードだ

さあ、いよいよゲームベーシック（GAME BASIC）に入るぞ。まず、画面でゲームベーシックを選択して——。

```
"GAME BASIC" ニ
シマスカ？
```



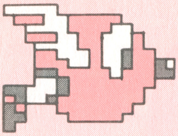




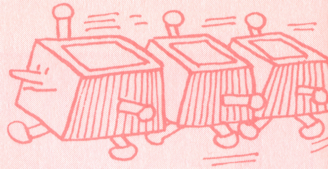


1がベーシックでプログラムを入力するための選択、2はゲームの背景や模様を作るためのプログラムだ。

まず、ベーシックの勉強からだから、1を押して——。スタートだ。



2.2 キーボードをマスターしよう



はじめはまちがえてもいいから どんどんキーを打ってみることだ

♣灰色キーは文字キー、赤色キーは特殊キーだ

プログラムを入力するにも、キーボードの使い方を知らなければおはなしにならない。

まず、キーボードの基本的な使い方からマスターしよう。

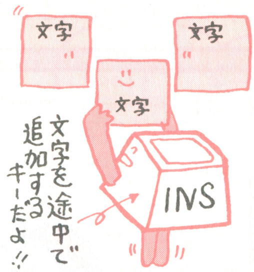
キーボードを見ると、赤色のキーと灰色のキーに分かれているね。

赤色のキーが特殊キー、灰色のキーが文字キーだ。

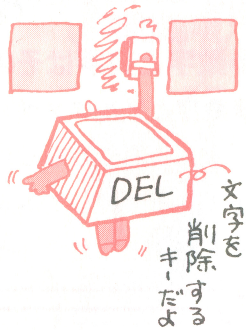
■ファミコン・キーボード



特殊キーというのは、そのキーにひとつの独立した機能が入っているキーのことだ。ファンクションキーや、文字を途中で追加するときに使う INS （インサート）キー、文字を削除する DEL （デリート）キーなども、特殊キーのひとつだ。



文字キーは、おそらくきみたちがいちばんたくさん使うキーだ。そのキーひとつだけでは意味はなさないけど、ベーシックの命令や数字を打ち込んで、コンピュータに伝える命令文を作りあげるのだからね。だから、この文字キーをスムースに打ち込めるようにすることがかんじんだ。



♣文字や記号を表示する位置のサインがカーソルだ

では、灰色のキーを見てみよう。ひとつのキーに、3～4個の文字や数字、記号が並んでいるね。

つまり、ひとつのキーにこれだけの意味が入っているということなんだ。

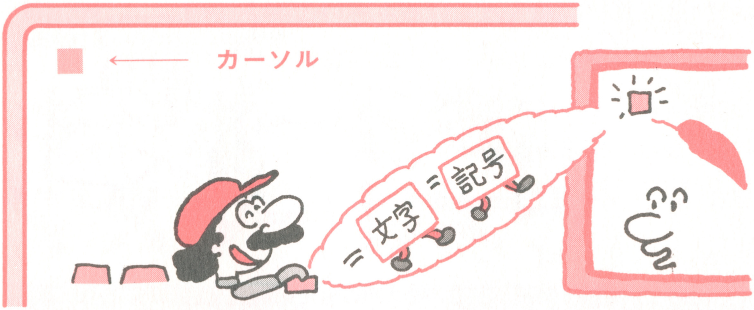
どうすれば、どの文字や記号が出てくるか。これをおぼえようというわけだ。

画面を見てみよう。チカチカと光る白い四角があるだろう？　これをカーソルというんだが、このカーソルの位置に、

●ポイント

ひとつのキーにいくつもの文字や記号があるから、その打ち分け方をおぼえよう。

文字や記号が表示されるんだ。



♣数字と英字はそのままキーを押すだけ

●数字とアルファベット

なんの操作もなしに、そのまま打てばいい。上段の数字キーを1～0まで押してみよう。次に、そのまま下の段の英字キーを押すんだ。どうだ？　うまく書けたかな？



● SHIFT キーを押せば記号が打てる

キーボードの下の左右に赤い大きめのキーが SHIFT キーだ。では、SHIFT キーを押しながら、数字キーを押してみよう。数字ではなく、記号が打てたはずだ。

SHIFT キーは、左右どちらのキーを使ってもいいよ

● カナは カナ キーで

カナ キーは、右側の SHIFT キーの上にある。このキーを押すと、カーソルが■から▬に変わる。▬に変わったら、カナが打てるよ、という印だ。変わらなかったら、もう一度押してみよう。あとは、キーボードの打ちたいカナを押せば、表示される。

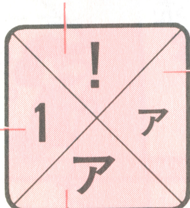


カナモードから、ふつうのモードにもどしたいときは、もう一度 カナ キーを押すともどるよ



記号は SHIFT キーを押して打つ

数字、アルファベット、「，」（コンマ）や、「・」（ピリオド）などは、そのまま打てばいい



カナの小文字や半だく音は、カナ キー＋ SHIFT キー

カナは カナ キーを押してから

●だく音は GRPH キーで

「ガ」や「ザ」などのにごる音を表示するときは、GRPH キーを押しながら、「カ」「サ」などを押せばいい。



●半だく音と小文字は SHIFT キーで

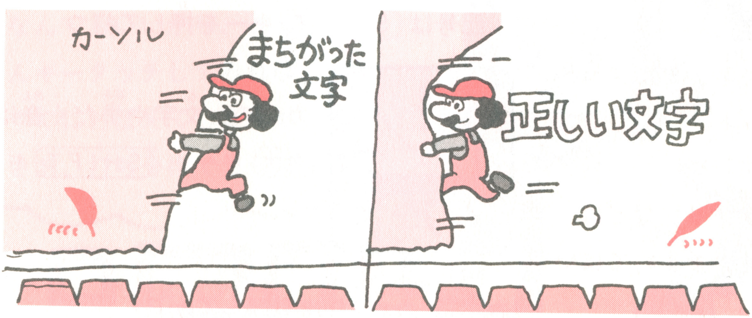
「パ」や「ポ」などの半だく音と、「キャ」「シャ」などの小文字は、SHIFT キーを押しながら、そのキーを押す。

♣打ちまちがえなんてヘッチャラだい

●まちがいを直すときはカーソルキー

まちがえて打ってしまったときに、正しい字に直したいときはどうするか？

キーボードの右がわに、▲ ◀ ▶ ▼ という4つのキーがあるね。これはカーソルキー、または方向移動キーといって、三角印（▲）は、文字やキャラクタが動く方向を示している。このキーを使って、カーソル（■または▃）をまちがえた文字の上に重ねるんだ。そして、正しい文字を打ち直す。これでオーケー。ちゃんと直ったはずだよ。




ヤマダ タロコ
カーソル
アッ、まちがえた!
ヤマダ タロコ
カーソルを動かしてから、ウのキーだ
ハイ、ヤマダ タロウになったぞ
ヤッタ!!
ヤマダ タロウ

●いらない文字を消したいときは DEL キー

打ちまちがえたり、よけいな文字を消したいときに使うのが、カーソルキーの上、いちばん右はしの DEL （デリート）キーだ。このキーを押すと、カーソルが1文字分左に移動して、左がわの文字を消していくよ。



●文字を追加したいなら INS キー

書きわすれた文字を入れたいときは、INS （インサート）キーだ。このキーを押すたびに、カーソルの文字が右に移動して、左がわに空白ができるよ。



■特殊キーについて

♣いろいろな機能があるから活用しよう

実際のプログラミングのときに役に立つキーたちだ。忘れたら、そのたびにこのページを開いて、操作しよう。

●一時停止の ESC キー



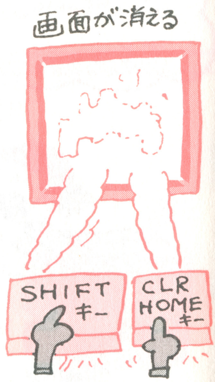
プログラムを見直すときに使うのが「LIST」というベーシック命令なのだが、このLIST命令を実行すると、画面にプログラムリストが表示される。長いリストが画面に出てきて、途中で止めて部分的に見直したいときには、この ESC （エスケープ）キーを押せば、そこで画面が止まる。また表示を続けたいときには、キーボードのどのキーでもいいから押すと、どんどん表示されていく。



●ストップの STOP キー

プログラムの実行を止めるキー。

●カーソルをもどしたり、画面のそうじをする CLR／HOME キー



CLR／HOME （クリアホーム）キーだけを押すと、カーソルが画面左上にもどる。ここをホームポジション（カーソルのもとの位置）というよ。 SHIFT キーを押しながら CLR／HOME キーを押すと、画面の文字がぜんぶ消えてしまう。

ベーシックのときのファンクションキー

ファミリーベーシックを選ぶときには［ハイ］、［イイエ］、うらないコーナーなら［コンニチハ］、ミュージックボードならドレミファソラシ、というふうに、このファンクションキーには、コーナーによって、いろいろな役わりがあったね。

ところで、“GAME BASIC”を選んだときには、またちがった役わりになるんだよ。

ゲームベーシックを動作させたときには、このファンクションキーには、よく使われるベーシック命令が入ってるんだ。

では、今なにが入っているか見てみよう。次のようにキー入力してごらん。

［K］［E］［Y］［L］［I］［S］［T］［↵］

（［↵］はリターンキーのこと）

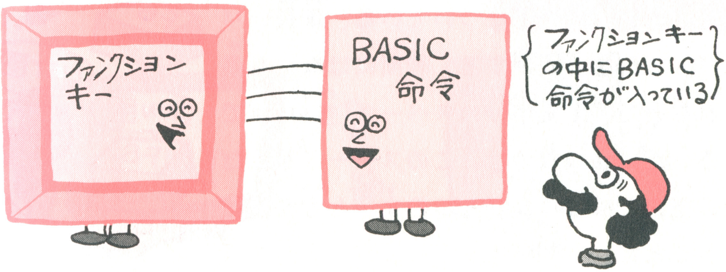
さて、画面に8個のファンクションキーの内容が表示されたかな。

ポイント

［RETURN］キーを“［↵］”という記号で表わすことがよくある。これから“［↵］”とあったら、［RETURN］キーを押すという意味だから、おぼえておいてね。

ポイント

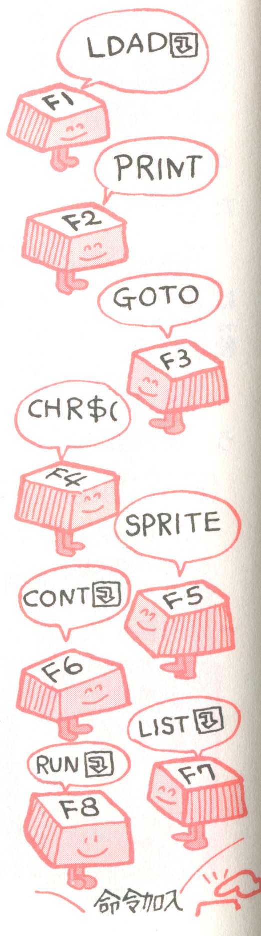
ファンクションキーになにが入っているかをおぼえておけば、正確で速いキー入力ができるよ。



F1………LOAD（M）
F2………PRINT
F3………GOTO
F4………CHR＄（
F5………SPRITE
F6………CONT（M）
F7………LIST（M）
F8………RUN（M）

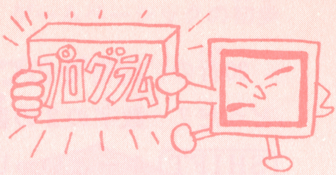


こんな英語のような文字を見ても、今はさっぱりわからないかもしれないけど、これらは、プログラムを組むときに、よく使う命令なんだ。もちろんキーボードから一字一字入力しても同じことだけど、ひとつのキーをポンと押すだけで、ひとつの命令が入力されてしまうのだから、便利だろう？　この命令の内容は、自分で変えることもできるんだよ。



●空白も１字分のスペースバー

べつにむずかしいキーじゃない。タイプライターの１文字送るときのバーと同じ。キーボードのいちばん下についている、細長いバーだ。これを１回押すと、画面にはなにも書かれないで、１文字分のスペース（空白）ができる。たいせつなのは、このスペースも、ただの空白ではなく、ひとつの文字と考えるということだ。



2.3 プログラミングにやっと挑戦だ！

プログラムがなければ コンピュータなんてただの箱なのだ

♣プログラムっていったいなんだろう

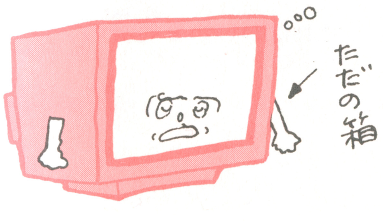
キーボード操作の基本を頭に置きながら、さっそく次のプログラムを入力してほしい。

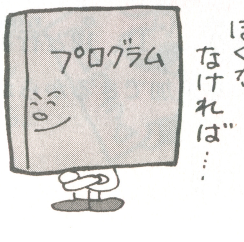
```
10  CLS
20  PRINT  "ヤマモト"
30  PRINT  "カワサキ"
40  END
```

ちゃんと入力できたかな？　念のために、まちがいやすい箇所を注意しておこう。

●ポイント

画面をきれいにしてから、ヤマモト、カワサキという文字を表示させるプログラムだ。

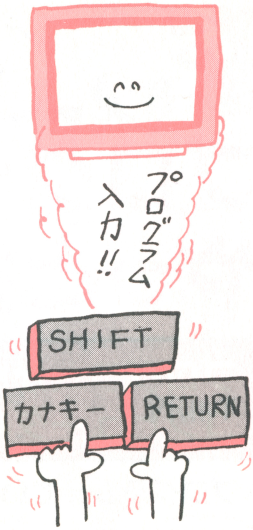




20行の〃(ダブルクォーテーション)はどうだ？　数字の2の書かれたキーを、SHIFT キーといっしょに押すのだったね。

カタカナ入力はだいじようぶかな？ カナ キーを押して、カーソルを■から▬にかえてから入力するんだぞ。

それからかんじんなのが RETURN キー。↵で表わしておくよ。リターンキーを押さないと、コンピュータはプログラムを記憶してくれない。1行入力するごとに、かならず押すこと。



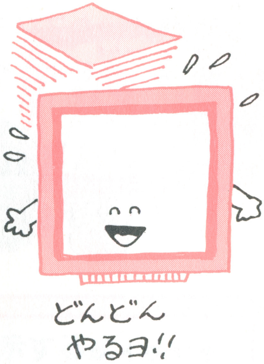
■行番号も知っておこう

プログラムの横に10、20、30、40と番号がふってあるだろう？　これは行番号といって、プログラムは、この行番号の順に実行される。つまり、仕事の順番をあらわす番号だ。

このプログラムでは、10きざみに行番号がつけられているけど、べつにきまりはない。100きざみにつけても、1、2、3とつけてもかまわないけど、この10きざみは便利なんだ。

たとえば、プログラムを途中に追加したいときには、10と20の間に、11、12と行番号をつければいい。9行分のプログラムを追加できるからね。

♣1行でも1000行でも同じりっぱなプログラム



たった4行だけど、これもりっぱなプログラムだよ。

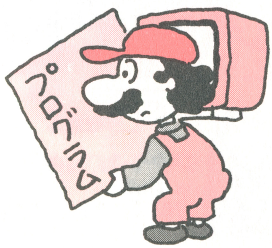
では、プログラムっていうのは、なんなのだ？

たとえば、10行を見てごらん。CLSというのは、「画面をきれいにしなさい」という命令で、クリアスクリーン命令という。では、20行は？　PRINT（プリント）というベーシック命令だ。40行は、「プログラムを終了せよ」というEND命令だ。



つまり、プログラムというのは、いろいろな命令を組み合わせて、コンピュータに目的の仕事をさせる命令書、ということだ。ここでは、「画面をきれいにしたあと、ヤマモト、カワサキという文字を画面に表示させる」という仕事だ。

♣わずか4行だけどいくつものベーシック命令



なんだ、こんな短いプログラム！　なんて、バカにしてはいけない。この4行のプログラムでも、いくつものベーシック命令を勉強することができるのだから。

まず、このプログラムを走らせてみよう。

■RUN（ラン）

♣入力したプログラムを実行させせよう

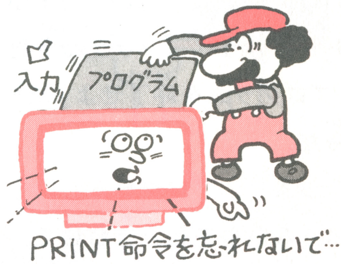
入力が終わったプログラムを走らせるには、次のように入力する。

R U N ↵

または、さっき勉強したファンクションキーの F8 を押してもいい。

画面から、さっき入力したプログラムの文字が消えて、プログラムにある「ヤマモト」「カワサキ」という文字がでてきたかな

```
ヤマモト
カワサキ
OK
■
```



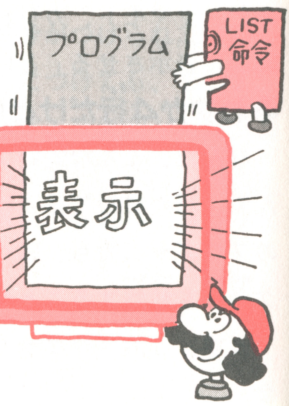
■LIST（リスト）

♣入力したプログラムをもう一度表示させせよう

プログラムを見直したり、修正したいときには、もう一度プログラムを画面に表示させなくてはならないね。この命令がLIST命令だ。

L I S T ↵

ファンクションキーの F7 、または省略形でL．でもオーケーだよ。



CLS（クリアスクリーン）

♣画面をきれいにそうじするよ

これは、実行してみればすぐわかる。プログラムの中に入れておけば、実行と同時に画面がそうじ（クリア）される。直接、キーボードから入力してもいい。

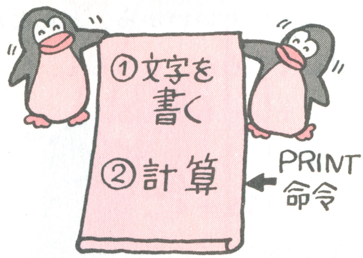


[C][L][S][↵]

PRINT（プリント）

♣文字も書くよ、計算した結果も表示するよ

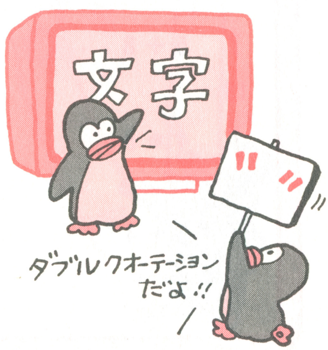
なぜ、画面に「ヤマモト」「カワサキ」と表示されたのか。もちろん、表示させる命令を与えたからだ。それが、20行、30行のPRINT命令だ。PRINT命令には、２つの働きがある。



●文字を表示するPRINT………〝 〟を忘れるな

まず、文字を表示するためのPRINT命令だ。このときに、表示したい文字を「〝 〟」（ダブルクォーテーション）で囲まないとエラーになるから、要注意だ。

ためしにきみの名まえでも入力してみよう。



[P][R][I][N][T][〝][ツ][ト][ム][〟][↵]

表示されたかな。PRINTのかわりに、?(クエッションマーク)、または省略形のP. でもいいよ。

これで、どんどん文字を表示させてみよう。

●計算するPRINT………" " はいらない

PRINT 命令のもうひとつの働きが、計算だ。

では、やってみよう。ここでは、?を使おう。

```
?  5+8 ↵
?  4*8 ↵
```

なんでもいいから、どんどん計算してごらん。もちろん、( )や、×、÷、−も使えるよ。

次に、

```
?  "5+4" ↵
```

と入力してごらん。

■計算記号

たし算……+

ひき算……−

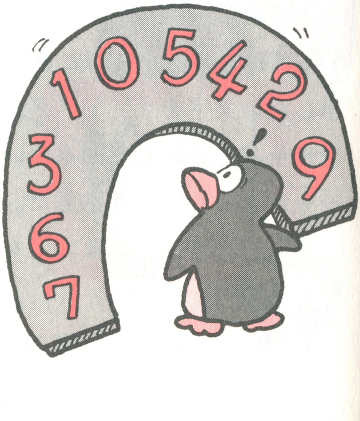
かけ算……*

わり算……/

カッコ……( )



```
?  5+8 ↵
13
?  4*8 ↵
32
? "5+4" ↵
5+4
```



〃　〃（ダブルクォーテーション）で囲むということは、「その中に入っている文字を表示しなさい」ということになるんだ。だから、「5＋4」という文字を表示してしまったというわけさ。

■ダイレクトモードとプログラムモード

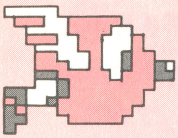
♣行番号をつけるかつけないかでちがうモード

さあ、どうだい。短くたって、きちんと命令で組み立てられているだろう。行番号をつけてプログラムを作り、コンピュータに仕事をさせる方法をプログラムモードという。プログラムは、行番号をつけてはじめて記憶されるんだ。

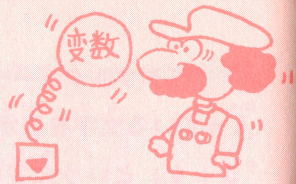
もうひとつ、プログラムを作らずに、直接キーボードから命令を入力して、リターンキーで実行させる方法がある。これをダイレクトモードという。

この方法、もうすでにきみも経験ずみだよ。CLSやPRINTのところで、いろいろためしたじゃないか。ダイレクトモードは、かんたんな仕事や確認には便利だよ。でも、ゲームやその他、大きな仕事をさせたいときには、プログラムを作らないとむりだね。





2.4 変数ってなに？



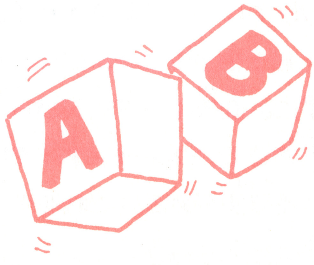
変数といってもヘンな数ではないぞ でも、変数を知らなきゃはじまらないよ

♣数値変数はいろいろな数を記憶させる魔法の箱

さあ、だんだんとプログラミングぽい勉強になってきたぞ。次のプログラムを入力してみよう。

■リスト1

```
10  A＝5
20  B＝8
30  C＝A＋B
40  PRINT  "C＝"；C
```



まず、LIST命令で、今入力したプログラムを画面に出して、まちがいがないかどうか確認してから、R U N ↵ でプログラムを走らせてみよう。どうやらこのプログラムは、計算処理をするプログラムらしいということぐらいはわかるな。画面の下に出た答えは——？



「C＝13」と出なかった人は、どこかに入力ミスがあるはずだよ。

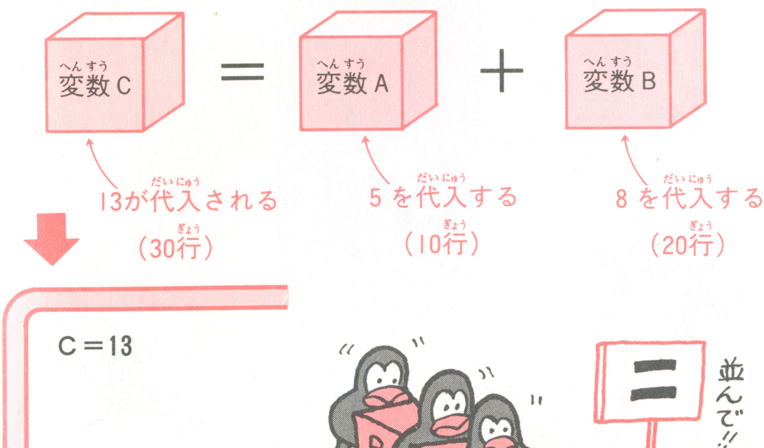
さて、プログラムの説明をしよう。ここで問題なのは、AとかBのアルファベットだ。これはいったいなんなのだ。

♣＝は「等しい」ではなく「代入する」の意味

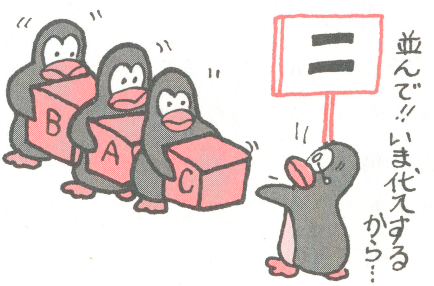
算数のかんたんな計算問題に似ているね。AとBの値をたした答えCは13だ。ただし、算数の計算では、＝（イコール記号）は「等しい」という意味をあらわすけれど、コンピュータでは「代入する、入れる」という意味なんだ。

では、どこに入れるのか。「変数」という箱に入れるんだ。このプログラムでは、A、B、Cが「変数」という箱にあたるんだ。

このプログラムを図であらわすと――――――。



変数Cに入った数を画面に表示する（40行）

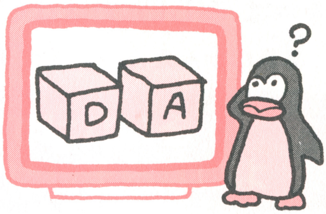


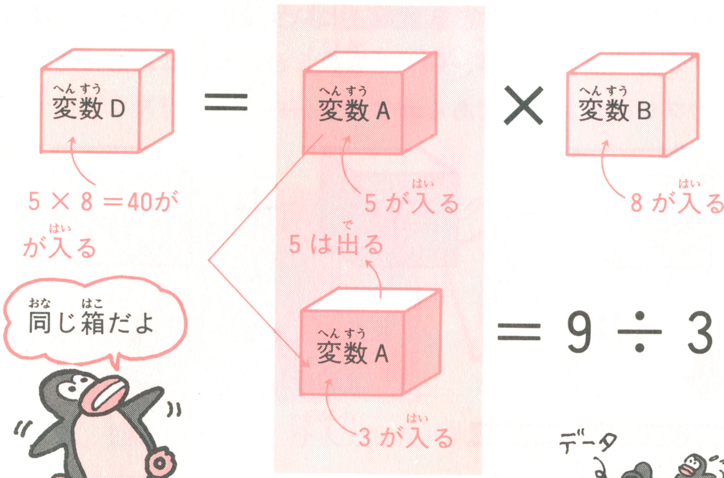
♣同じ箱にデータが入ったり出たり

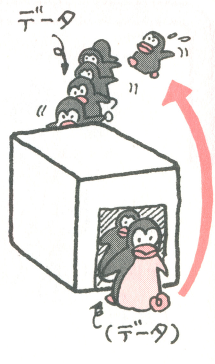
この変数という箱の性質を、さらによく理解するために、もうひとつプログラムをあげておこう。

■リスト2

```
50  A＝5：B＝8：C＝9
60  D＝A＊B：A＝C／3
70  PRINT  D，A
```



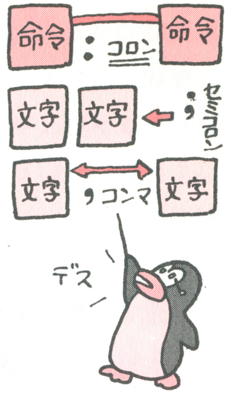




60行の「A＝C／3」を「5＝9／3」なんてやった人はいないだろうね？　ここは、Aの箱にC／3の値を入れるという意味だけど、Aという箱はひとつしかない。そうすると、前にAの箱に入っていた5は、どうなってしまったのだろう。

これが変数の性質のたいせつなところだよ。同じ変数に新しいデータを入れると、古いデータのうえに新しいデータがコピーされるので、古いデータは、自然に消されてしまう。

Aの箱を使ってもっとデータを与えてやれば、Aの値はどんどん変わる。変数は、変な数じゃなくて、変化する数ということができるね。



■：と；と，

●命令と命令をつなぐのが「：」（コロン）

これは、プログラム整理のコツともいえるのだけど、ひとつの行番号の中に、複数の命令をつづけて書き込むときに使う。たとえば、リスト2では、「60　D＝A＊B」「70　A＝C／3」としても同じことだけど、この2つの命令を「：」でつないで、ひとつの行にまとめているんだ。これをマルチステートメントという。

●文字と文字をつづけて画面に表示させるのが「；」(セミコロン)

リスト1で、「C＝13」と画面に表示されたはずだ。「；」は、文字と文字をつづけて画面に表示させたいときに使う。

●文字と文字をはなして画面に表示させるのが「，」(コンマ)

リスト2で、Dの値とAの値は、はなれたところで画面表示されたね。これは、PRINT文で画面表示させるときに、AとDを「，」で区切ったからだ。

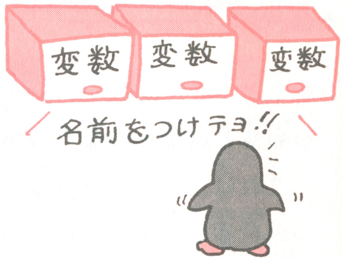
■変数の名まえ

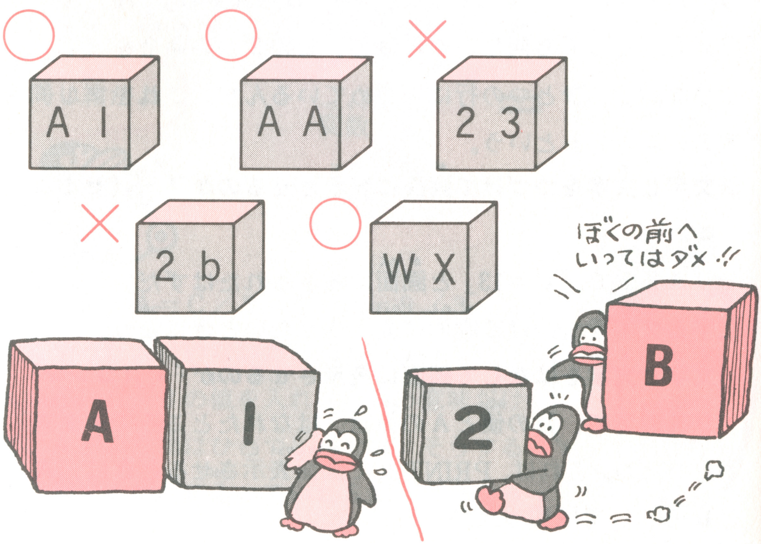
変数にも名まえをつけてやらないと、コンピュータが判断できない。ただし、いくつかの約束ごとがある。

1．アルファベットのA～Z、数字の0～9までを使って、2字以内でつけること。
2．最初の1文字はアルファベットであること。
3．数字だけの変数名はつけられない。

●ポイント

この変数の名まえを「変数名」、変数に代入される値を「変数値」というよ。





♣文字変数は$マークつきで文字を入れる箱

データは数値ばかりとは限らない。文字のデータだってあるよ。たとえば、ゲームなら、「アタリ！」とか「オワリ」なんていうのも、文字データで、プログラムのどこかに入っているはずだよ。

だけど、数値と文字とはいっしょには入れられない。文字には文字専用の変数が必要だ。これを「文字変数」または「ストリング変数」というんだ。文字変数にもまた、約束ごとがある。

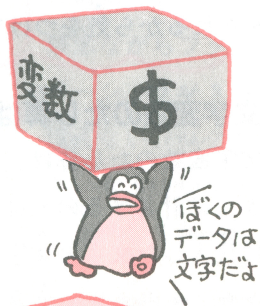
1．文字変数は、変数名のあとに＄（ドルマーク）をつけて、数値変数と区別する。

2．文字変数に入れるデータは、かならず〝　〟(ダブルクォーテーション）で囲むこと。

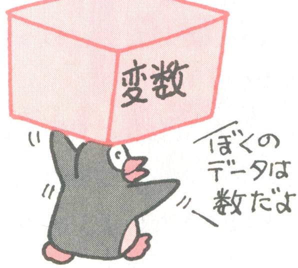


■リスト3

```
25　A＄＝〝ストリング〟
26　B＄＝〝ヘンスウ〟
45　PRINT　A＄＋B＄
```



リスト3をキー入力したあとで、L I S T ↵ で、画面にリストを出してみよう。アレレ!?　リスト1とリスト2がつながって、今入力したプログラムが、リスト1の間に入っているよ。10きざみに行番号をつけておくと便利だとい

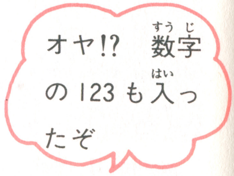


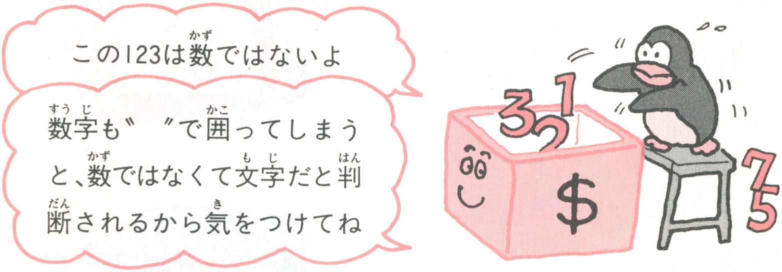
った意味がわかったね。行番号は、あとから書いても、数の小さい順にちゃんと並べてくれるんだ。さあ、[R][U][N][↵]だ。

♣文字変数に代入された数字は数ではなくて文字列だ

■リスト4

```
80   A$＝"アタリ":B$＝"123"
90   A$＝A$＋B$
100  PRINT  A$
```





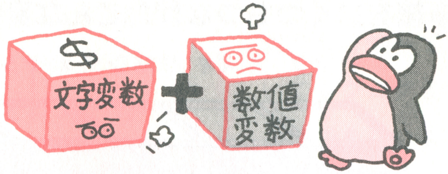
♣文字変数のたし算は文字と文字をつなぐこと

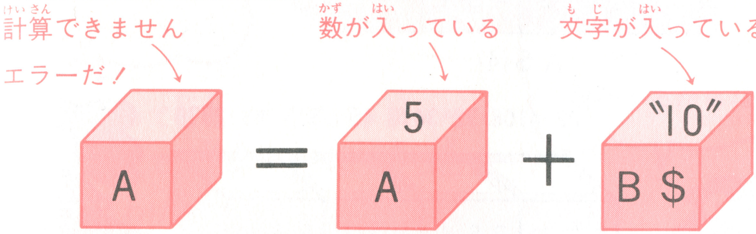
文字変数のたし算は、数値変数のように計算するのではなく、変数に代入した文字と文字をつなぐことだ。もちろん、文字だから、ひいたり、かけたり、わったりはできない。



また、数値変数と文字変数は別のものだから、2つを組み合わせることはできないよ。次のプログラムは、だめな例だぞ。

```
10　A＝5
20　B＄＝"10"
30　A＝A＋B＄
```





RENUM（リナンバー）

♣プログラムの行番号を整理しておこうネ！

変数の勉強のところで、あとから半端な行番号を入力したりしたので、なんだかバラバラな行番号になってしまったね。よし、次のように入力しよう。

R E N U M ↵

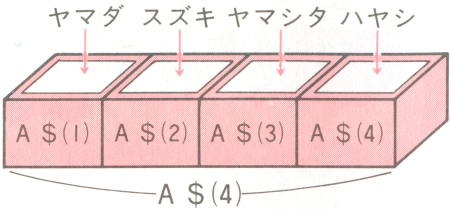
これは、RENUM（リナンバー）命令といって、プログラムの行番号を整理してくれるんだよ。画面下に、OK と出ていれば、

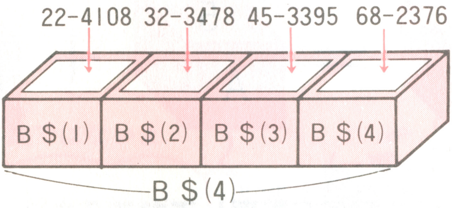
L I S T ↵

プログラムの行番号は、10番きざみに整理されているだろう。この行番号の間にも、またプログラムを追加できる。便利だね。



♣箱をいくつもつなげたこんな変数もあるよ



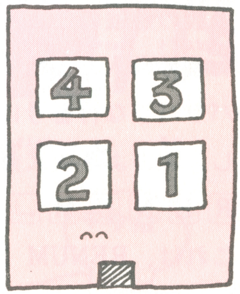


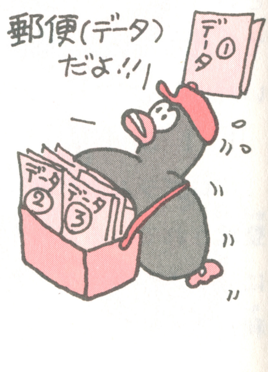


「なんだ、この変数は。アパートみたいだな」。この変数は、二次元配列の変数というのだけど、同じ変数名でいくつものデータを入れることができる便利な変数だ。そのかわり、同じ名まえの変数がいくつあるのかをきちんときめてやらなければいけないよ。

この二次元配列は、同じ種類のデータを入れておくのに便利なんだ。住所録とか電話番号の管理などに使われることが多いみたいだね。

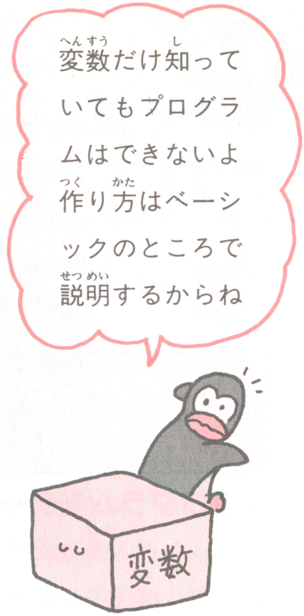
たとえば図のように、A＄には電話番号をわけて入れてみた。ここでは4人の名まえと電話番号を入れたから、A＄に





は4つの箱がある。つまり、A＄(4)。同じように、B＄(4)ときめてやる。そして、「箱の数を決めましたよ」とコンピュータに知らせるために、DIM（ディメンション）という命令を前につけておく。これを忘れるとダメだよ。

こうやって整理しておけば、調べたい人の名まえで、電話番号をすぐ呼び出せるというわけさ。



♣変数の内容を確認したかったら

「アレ!? 変数Aになにを入れたのかな」変数の中を調べたいときに便利なのが、STOP（ストップ）命令だ。

```
10　A＝15
20　B＝A＋21
30　　:
40　　:
```

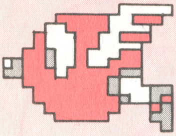
25　STOP　と入れておく

STOP命令のところでプログラムの実行が一時停止されるから、ダイレクトに「？B」とキー入力。Bの変数には36が入っていると教えてくれるんだ。プログラムを手直しするときに役立つぞ。

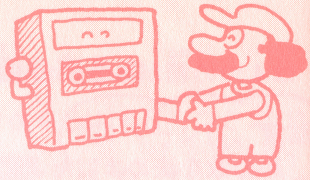
そのままつづけたければ、こんどはCONT（コンティニュー）命令だ。ダイレクトに入力して、次の行から実行だ。







2.5 プログラムのセーブとロード



プログラムが消えないように保存したり 保存したプログラムを読み出したり

♣プログラムをテープにとっておこう

せっかく入力したゲームプログラムも、電源を切れば、あとかたもなく消えてしまうのがパソコンだ。

でも、安心して。プログラムを残しておく方法があるんだよ。ちゃんととってさえおけば、プログラムは何回でもくり返して使えるからね。

きみはカセットテープレコーダを持っているかい？　もし、持っていなければ、ファミコン専用のデータレコーダ（HVC−008）というのがあるから、手に入れるといい。専用のデータレコーダの方が、ミスは少ないよ。



さあ、キーボードに接続して、テープをセットしておこうね。

SAVE（セーブ）

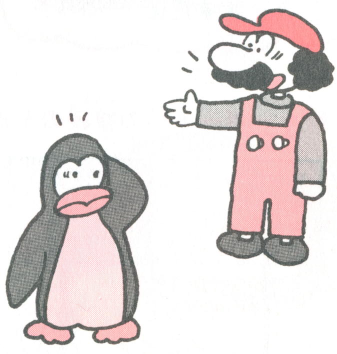
♣名まえをつけてプログラムを保存しよう

「セーブ」というのは「保存する」という意味のことばだよ。セーブする前に、プログラムにも名まえをつけてやらないと、あとで呼び出して使うときにこまる。プログラム名は、16文字以内でつけ、〝　〟（ダブルクォーテーション）で囲うというきまりをおぼえておこう。

プログラムをセーブする命令は、SAVE（セーブ）命令だ。

次の順番を守ってセーブすれば、プログラムはテープに移るよ。ここでは、プログラム名を〝ケイサン〟としておこう。

1．ダイレクトモードでキー入力。**SAVE〝ケイサン〟**
2．テープレコーダの録音スタート
3．　（リターンキー）を押す
4．画面に「WRITING ケイサン」と表示される
5．OKと出たらセーブ完了



♣ちゃんとセーブされたかカ〜クニン！

なれないうちは、セーブミスもあるだろう。テープにとったあとも、すぐにパソコンの電源を切らずに、念のためちゃんとテー

プに入っているかどうかを確認することがたいせつだ。確認する方法は、次のとおりだ。

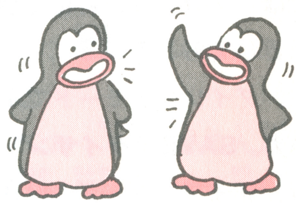
1．テープを巻きもどす
2．**LOAD？**とキー入力
3．⏎を押す
4．テープスタート
5．画面に「LOADING　ケイサン」と出る
6．OKと出れば、確認完了

●ポイント

もしもセーブがうまくできなかったら、テープレコーダのボリュームを少し高めにセットしてごらん。

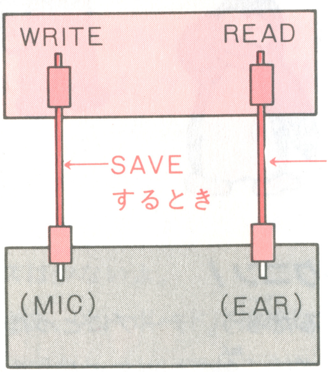
LOADING　ケイサンって？

〝ケイサン〞というプログラムをみつけましたよ、という意味だよ



■キーボードへの接続のしかた

キーボード
裏面





データレコーダ
（ラジカセの場合も同じ）

LOAD（ロード）

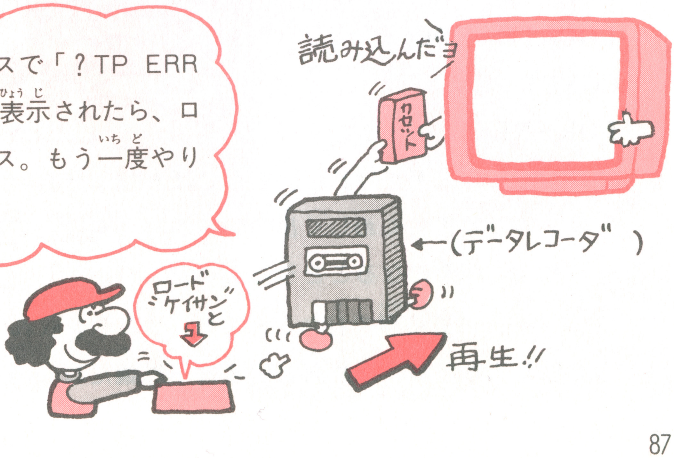
♣しまっておいたプログラムを引き出して使う

「ロード」。また聞きなれないことばだ。「ロード」というのはLOAD（ロード）命令で、「プログラムをコンピュータ本体に読み込む」という意味だよ。では、いまセーブしたプログラムを、さっそくロードしてみよう。



1．プログラムをセーブしたテープをカセットにセットする
2．**LOAD "ケイサン"** とキー入力する
3．↵ を押す
4．テープレコーダの再生ボタンを押す
5．画面に「LOADING　ケイサン」と表示される
6．OKと出れば完了

入力ミスで「?TP ERROR」と表示されたら、ロードミス。もう一度やり直し



かんたんなことはいいことだ

♣ベーシック命令の省略形を活用しよう

おぼえてしまえば楽になる
リストにとれば正式な語に変わる

ベーシック命令語は、だいたいかんたんな英語で表現されているから、その意味はおぼえやすいね。でも、たとえば、PRINTとかGOTOとか、キーボードから打ちこんでいくのが、めんどうだなどというブショウモノはいないかな？

そんなブショウモノたちに便利なのが、命令語の省略形っていうわけだ。よく使われる命令語の省略形を教えておくから、利用してね。矢印のうしろが省略形だ。

PRINT→？またはP.

GOTO→G.

LIST→L.

LOCATE→LOC.

CGSET→CG.

FOR～NEXT→F.～N.



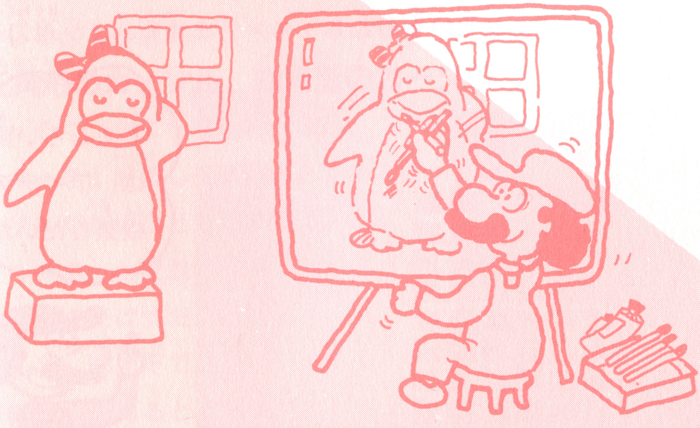
これはほんの一例だけれど、さがせばまだまだあるよ。ベーシックの参考書にくわしく出ているから、おぼえてしまえば、これからが楽になることうけあいだ。

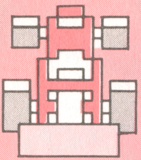
プログラムに省略形で打ちこんでも、リストを取れば正式な形で出てくるから、不思議といえば不思議だね。

PART 3

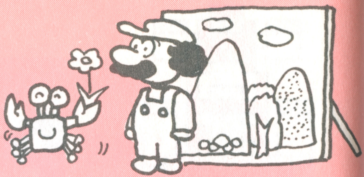
BG グラフィック

BG　グラフィック（GRAPHIC）は、きれいな絵の具ベタベタの背景画づくりの道具だよ。ゲームだって、背景の絵がきれいにできれば、それだけで楽しいものね。





3.1 背景の絵を作ろう



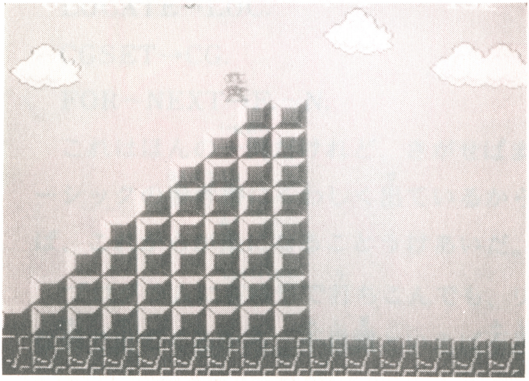
キャラクタを自由自在に組み合せて いろいろな色でいろいろな図形作り

◆キャラクタという図形のもとを選んで

BGはバックグラウンド（Back Ground）の略、グラフィック（GRAPHIC）は絵や模様の意味だ。

ゲーム中に飛行機が飛んでいたり、船が動いていたりしても、空や雲や海があるのとないのでは、ゲームのおもしろさがだんぜんちがうだろう？

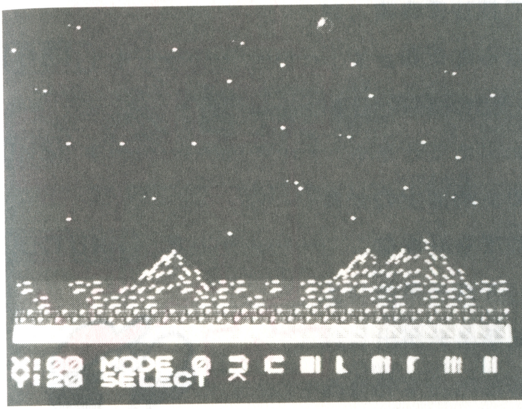
つまり、BG グラフィックとは、動かない背景の絵を別のところで作って、ゲームそのものとドッキングさせて、より迫力ある





ゲームや画面にしようというものだ。

ベーシックをはじめる前の画面にもどって、“BG GRAPHIC”を選択するところからだ。





この“BG GRAPHIC”を選択すると、画面がかわって、下のほうにいろいろな図形のようなもようが出てきたね？　そこから選べばいいだけだから、やさしいだろう？

そして、この図形のことを「キャラクタ」というんだ。

◆三角形のカーソル、四角形のカーソル

さっそく、なにか描いてみよう。スペースキーを１回押してごらん。すると、カーソルが右にひとつ移動して、前にカーソルのあった位置に、キャラクタがひとつ出てきたはずだ。

そこで、画面下で点滅している三角形のカーソルのあるところを見てごらん。

わかっただろう。この三角形のカーソ

> **●ポイント**
> 三角のキャラクタ・マーカーで選んだキャラクタを、カーソルの位置に移動して表示させるのだ。

ル位置にあるキャラクタを、画面上の四角いカーソルの位置に移すだけなんだ。同じように、スペースキーを9回つづけて押してみよう。同じ図形が10個ならんだね。三角形のカーソルは「キャラクタ・マーカー」とよぶよ。

カーソル移動キーで、画面上の四角いカーソルをどこでも好きなところに移動させて、スペースキーを押すと、またキャラクタが出たね。

こうして、三角カーソルと四角カーソルを見ながら、図形を組み合わせていくわけだ。(ESC)キーを押してから(STOP)キーを押すとGAME BASICモードの画面に戻るよ。



◆104種類のキャラクタに色もいろいろ

キャラクタは、ぜんぶで8×13の104種類も用意されているから、アイデアしだいで、本格的な背景画が描けるはずだ。はじめは、とにかく、いろいろなキャラクタを組み合わせて、どんどん絵を作ってみることだ。

その前に、次のキーを頭にしっかりと入れておかないと、自由自在に背景画を描けないぞ。

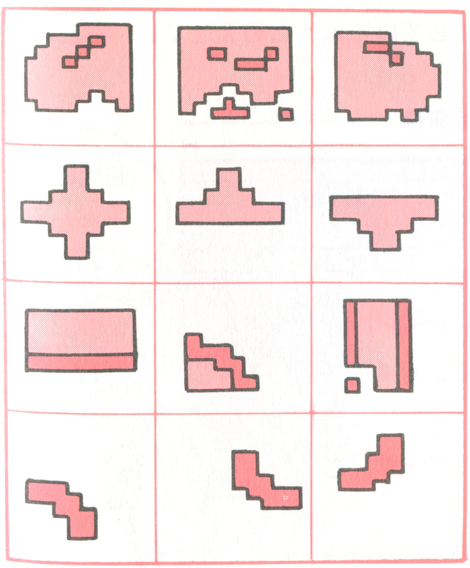
(CLR／HOME) キー…… このキーを押すと、A～Mまでのまとまったグループごとに順に図形がかわる。(SHIFT)キーといっしょに押せば、前にもどる。

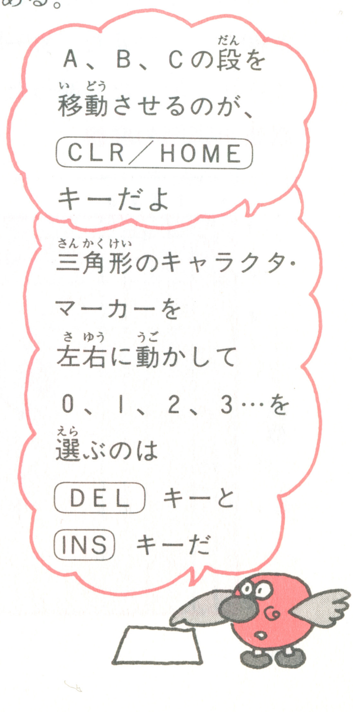
(DEL) キー…… キャラクタ・マーカー（カーソル）は右へ移動させ、キャラクタを選ぶ。

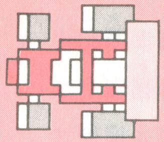
(INS) キー…… キャラクタ・マーカーを、左へ移動させ、キャラクタを選ぶ。

(RETURN) キー…… このキーを1回押すと、MODEのあとに表示されている数字がかわる。これは、配色を選ぶキーで、0～3までの合計4色がある。

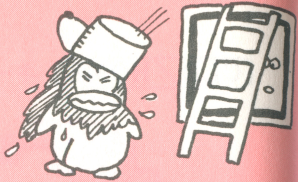
■BGグラフィックのキャラクタ







3.2 どの位置にどんな色で表示する？

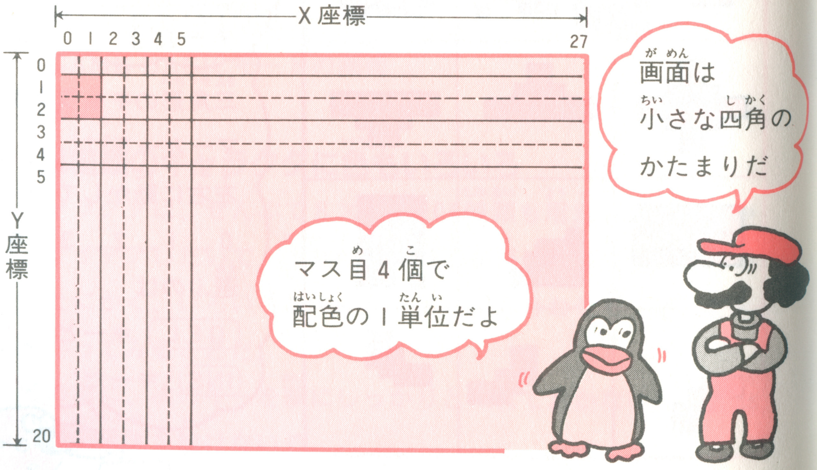


座標なんていうとむずかしそうだけど表示する場所はどこかということだ

◆XとYは表示する場所の座標だよ

画面左下に、XとYという文字があるね。はじめはX：00、Y：00だったはずだ。カーソルを移動させると、このXとYのうしろの値が増えたり減ったりしているね。このXとYは、画面の中

■表示画面の座標



の座標位置を表示しているんだ。

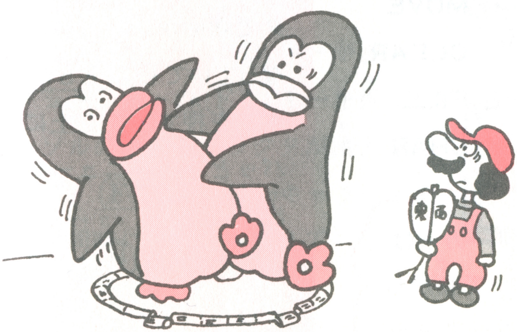
◆マス目4個が同じ配色の単位

キャラクタが描ける範囲は、X座標で00から27までの計28、Y座標は、00から20までの計21の座標内だ。合計すると、588のマス目にキャラクタが描けるんだ。

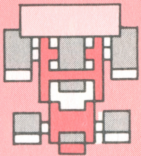
それから色つけのことだけど、図の中に■で示したところがあるね。ここには4つのマス目が集まっている。この単位をカラーエリアといって、このカラーエリアには、1つの配色番号しか決められない。つまり、この中で2色も3色もは使えないのだ。

ためしに、たてに、色をかえてキャラクタをならべてみてごらん。上のキャラクタの色が、今描いた色にかわってしまうことがあるはずだよ。配色をするときには、気をつけてね。

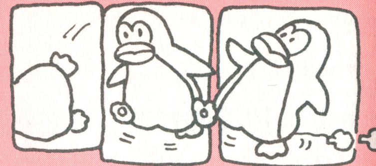
ただし、いちばん上の段だけは、横2つのマス目がエリアになっているよ。



たとえば1つのエリアの中に配色番号0のキャラクタをおいたあと、それと同じエリアに配色番号3のキャラクタをセットしたら、前にあったキャラクタも配色番号が3になってしまう。



3.3 背景を描くテクニック



モードのメニューいろいろ
コピーしたり動かしたり、いろいろだ

◆マーカーを移動させて、さてどのモードを選ぼうか？

ESC キーを押してごらん。画面上の左上に文字が出てきたはずだ。これは、背景画を描くためのテクニックに必要な画面モードのメニューいろいろだ。

■BGグラフィック6つのメニュー



```
> SELECT
  COPY
  MOVE
  CLEAR
  FILE
  CHAR
```

＞のマーカーは▲▼で動かしてスペースキーで選ぶんだよ

●ポイント

いろいろなキャラクタで画面がいっぱいになっていても、ESC キーを押すと、左上にパッとこのメニューが出てくる。

この部分以外は、描き込んだキャラクタはそのままだよ。

■セレクト（SELECT）モード

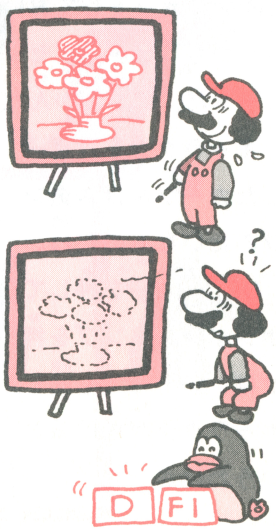
◆BGグラフィックを選んだらこの画面になっている

モードというのは、それぞれの機能を使うための画面の状態、とでもいうのかな。たとえば、カナ キーを押して、カタカナが入力できる状態をカナモードというし、絵を描ける状態に切りかえたら、グラフィックモードというしね。

BG グラフィックを選んだときに、すぐに出てくる画面はセレクト（SELECT）モードにセットされている。このモードは、画面を表示させる機能のことだ。他のいろいろなモードで、描いたり消したり、くふうしたりして作った画面を、最終的にひとつの作品として表示させるのも、このモードだ。使い方は、いままで説明してきたとおりだ。

◆ D か F1 キーを押せばキャラクタが消える

また、このモードは、キャラクタを表示できるだけでなく、逆にキャラクタを消すこともできるから、おぼえておこう。



アルファベットの D キーか、ファンクションキーの F1 キーを押すと、カーソルの中のキャラクタが消える。ただし、F1 キーは、キャラクタを消すと同時に、配色も変えてしまうから、要注意だ。

さて、これからSELECTモード以外の5つのメニューを説明していこう。

■コピー（COPY）モード

◆いくつも使いたいキャラクタをコピーする

画面の左上に6つのメニューを表示させたまま、こんどはカーソル移動キーで“COPY”モードを選んで、スペースキーを押そう。

画面はどうなったかな？　6つのメニューは消えて、画面はセレクトモードで描き込んだときの画面にもどったはずだ。そして、画面下には、「コピーモードになったよ」というしるしに、「COPY」と表示されているね。

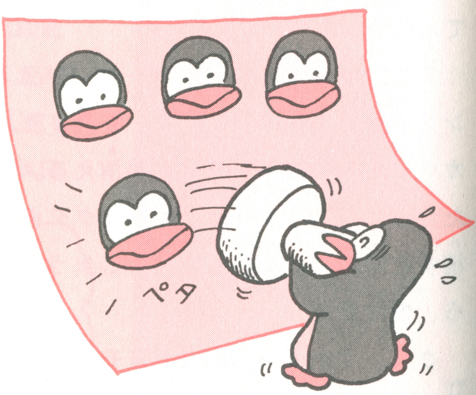
このコピーモードは、セレクトモードで画面にセットしておいたキャラクタを、いくつもコピーしてセットできる便利なモードだ。でも、かんたんなテクニックなんだよ。

●ポイント

モードが変わるとカーソルキーと ESC キー以外は、キーの使いかたが変わるので注意しよう。

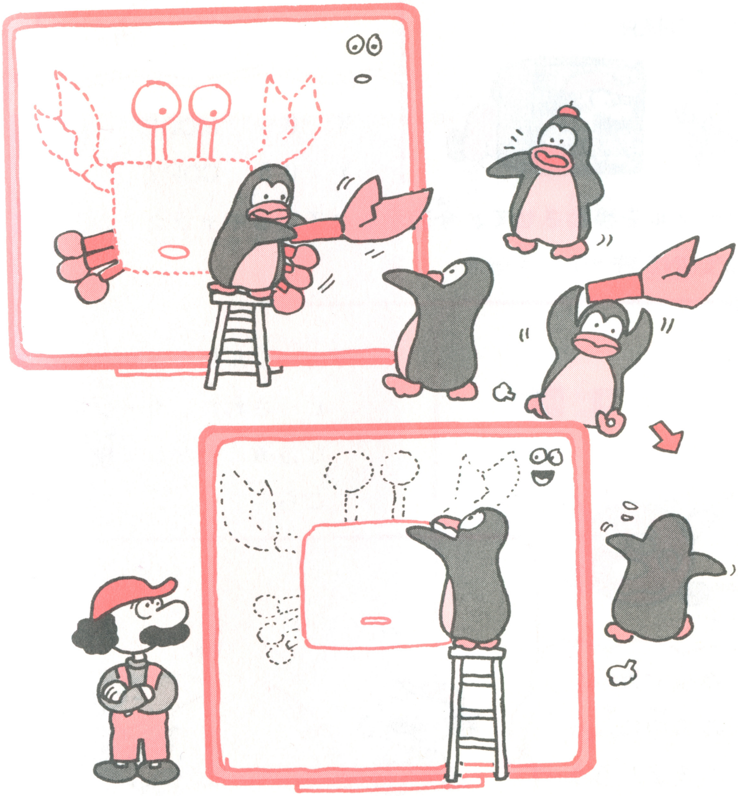
ESC キーは、6つのメニューを表示して、目的の機能を選ぶときに押すのだったね。

キャラクタをコピーしたカーソルはスタンプみたいなものだね。DEL キーを押して、ポンポンスタンプを押せば、いくらでも同じキャラクタが描けるというわけさ。



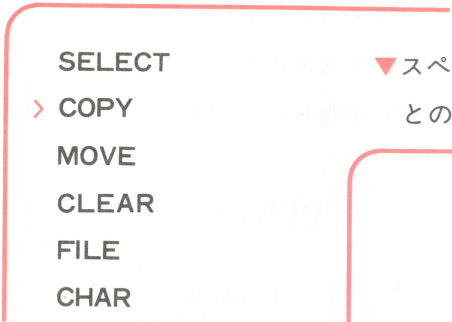
いったんセレクトモードで描き込んだキャラクタの中で、いくつも使いたいキャラクタがあったら、その上にカーソルを重ねて INS キーを押す。すると、カーソルにキャラクタがコピーされる。もちろん、もとのキャラクタはそのままの位置にいきてるよ。そのカーソルを描きたい位置に移動させて DEL キーを押すと、そのキャラクタが画面に表示される。

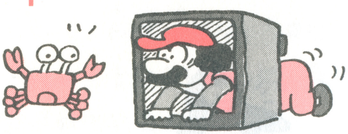
他のキャラクタも INS キーと DEL キーでためしてみよう。



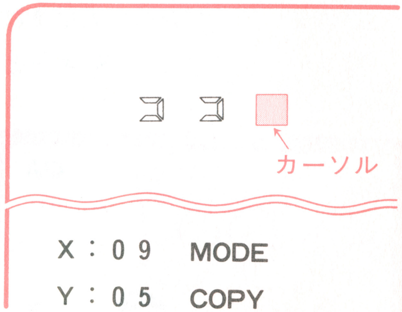
■コピーしてみよう

▼ESCキーを押してから、COPYモードを選ぶ。

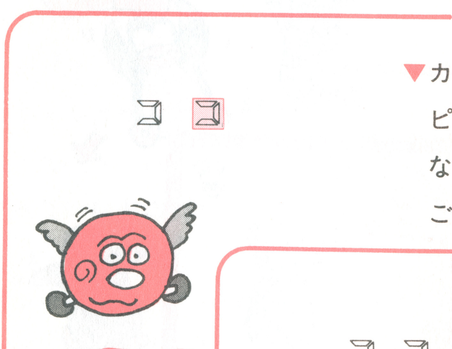




▼スペースキーを押すと、もとの画面にもどる。

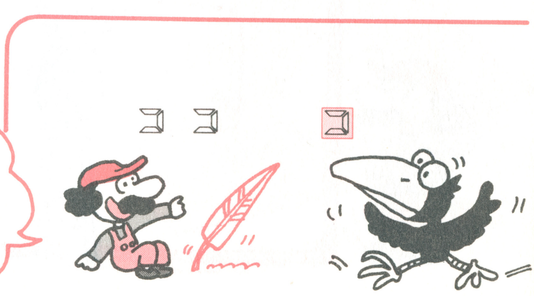


▼カーソルを移動させて、今表示されているキャラクタの上に重ねてINSキー。

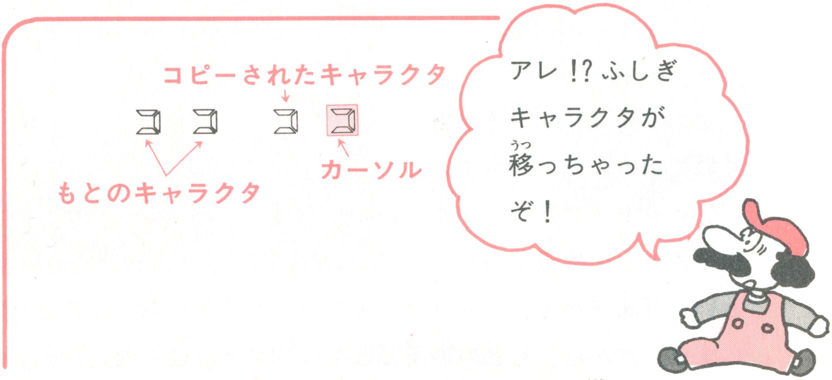


▼カーソルにキャラクタがコピーされた。カーソルを別な位置に動かして確認してごらん。

ネッ！カーソルの中に入ったままだろう



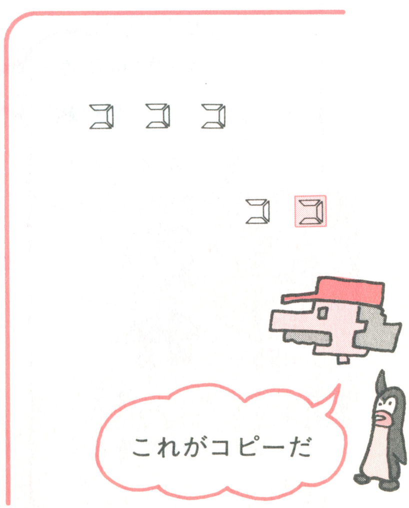
▼では、まだなにも描かれていないところにカーソルを移動させて［DEL］キーを押してからカーソルを移動させてみよう。



●**ポイント**

おもしろい使いかたもある。描いたキャラクタを消すには、セレクトモードで［D］キーを押すっていったよね。コピーモードで消すときは、カーソルの中に空白をコピーしておくんだ。カラの状態で［INS］キーを押すんだよ。それを消したいキャラクタの上に重ねて［DEL］キーを押す。空白をコピーするということは、そう、消えることということだ。頭は使いようだな。

▼カーソルを移動させると、キャラクタが残って、まだカーソルの中にはキャラクタが入っているままだ。



■ムーブ（MOVE）モード

◆キャラクタをどこにでも引っ越しさせる

同じように［ESC］キーでメニューを出して、MOVEを選んでスペースキーだ。

このムーブ（MOVE）モードは、コピー（COPY）モードとよく似ているようで、ちょっとちがう。

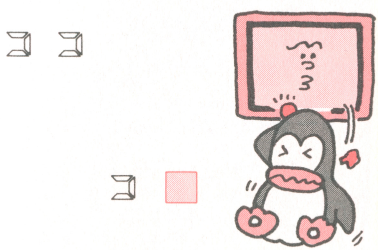
コピーモードはコピーするわけだから、もとのキャラクタはそのままもとの位置に残っている。「MOVE」というのは「動かす」という意味だから、キャラクタそのものを別の位置に移動させてしまう。だから、もとの場所にはキャラクタは残らない。



◀まず移動させたいキャラクタの上にカーソルを重ねて［INS］キーを押す。



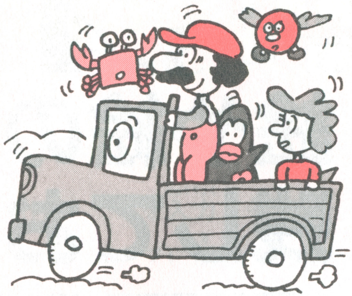
▶移しかえたい位置までカーソルを移動させて［DEL］キーを押す。



移動させた位置にキャラクタが残り、カーソルの中はからっぽに

●ポイント

ムーブ（MOVE）モードでのカーソルの役目は、引っ越しのトラックや荷物を運ぶ台車のようなものだね。目的地に届けたら中味はからっぽになるよ。

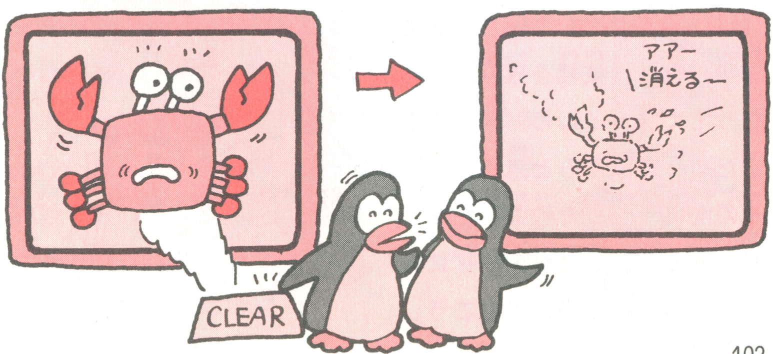


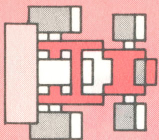
■クリア（CLEAR）モード

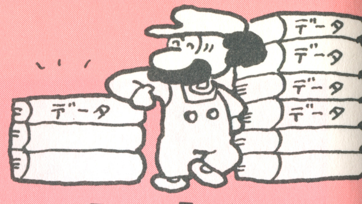
◆必要な絵までうっかり消さないでね

クリア(CLEAR)モードは、慎重に使うこと。マーカーを“CLEAR”にあわせて、スペースキーを押したとたん、ぜんぶの絵が消えてしまい、再生することはできないのだから。うっかり使うと命とりだぞ。

一部分だけ消したいときは、セレクト（SELECT）モードの D キーを押して消せば、安心だ。今描いていた画面が、ほんとうにぜんぶいらないのかどうかをよく考えてから、このモードを選ぶんだよ。







3.4 絵や絵のデータを残しておこう

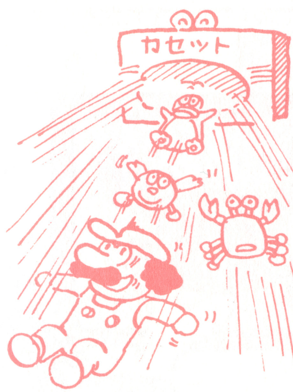
**せっかく苦心して作った絵なのだから
あとでまた使えるようにしておきたい**

■ファイル（FILE）モード

◆画面はきれいにしたいし絵も残したい

画面に描いた絵は、いったん電源を切ると消えてなくなってしまうし、新しい絵を描きたいときは、どうしても画面をぜんぶ消してきれいにしなければならない。でも、それまでの絵も残しておきたいときに使うのがファイル（FILE）モードだ。

テープレコーダの接続とセーブ（SAVE）命令、ロード（LOAD）命令については2―5で説明したね。接続を確認したら、さっそくファイルモードを選んでためしてみようと思うけど、その前に、今まで勉強したモードを使って、なにかテープに入れるための絵を描いておこう。



◆キーボードからSまたはLを入力

▼FILEモードにすると、画面の絵が消えて、次の画面になる。

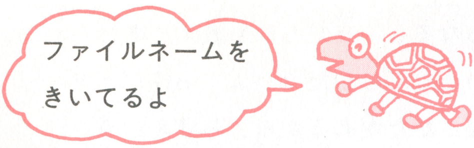
SAVE（S），LOAD（L）?■

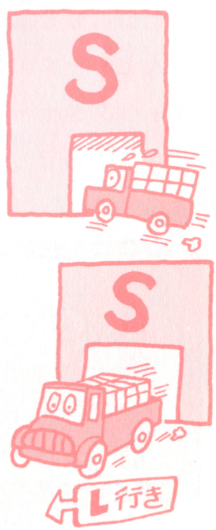


ここで、キーボードから、SかLを入力する。テープに記録するならS、テープから呼び出すならLだね。

SAVE（S），LOAD（L）?S
FILE NAME "■

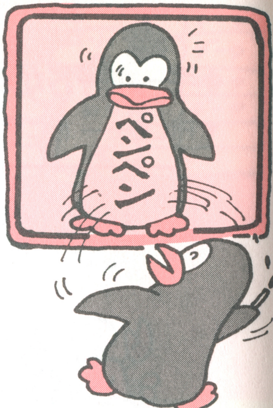
SAVEのSだ





保存するならS、呼び出すならLをキー入力すると、次にファイルネームを聞いてくる。ファイルネームも前に説明したとおりだ。「ウチュウ」なんて、かってに好きなタイトルをつけてもいいよ。

あとは、前に説明したのと同じ操作を行なえばいい。テープにSAVEする時間は約1分30秒くらいかな。途中で中止して、画面表示にもどりたいときは、STOP キーを押せばいいよ。



■キャラクタ（CHAR）モード

◆グラフィック画面にキャラクタを書く

キャラクタ（CHAR）モードにすると、グラフィックの画面に、キーボード上の文字や数字、記号を書くことができる。このモードはかんたんだから、自分でいろいろためしてみよう。

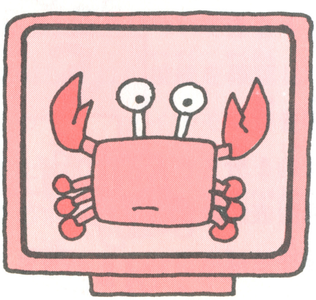
■背景データの記録

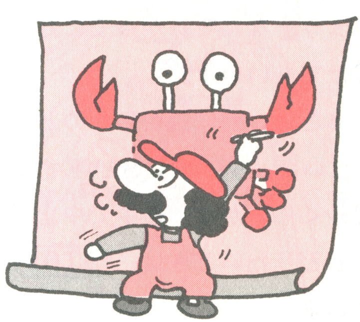
◆キャラクタテーブルBを使ってコレクション

画面に描いた絵をその場かぎりで消してしまわずに、とっておけば、りっぱなコレクションができるぞ。テープにとっておくのもひとつの方法だ。さらに、どうやって作ったのかを紙に記録しておけば、参考にもなるし、友だちにも教えてあげられる。

ただ紙に絵を描いたってしかたないよ。BGグラフィックにはちゃんとデータの書きかたがあるから、おぼえよう。

キャラクタテーブルBを使うんだ。A～Hまでのアルファベットと0～7までの数字は、このデータを作るためのものだったんだよ。





◆X座標、Y座標をきっちり決めて

紙に記録するときも、きちんと座標を設定しないと、ひと目でどんな絵になるかわからないぞ。その座標に、データを記入していくんだ。データは、キャラクタグループ（A～H）、番号0～7）、配色の3つの要素を入れて作るよ。

A 2 3
配色
番号
グループ

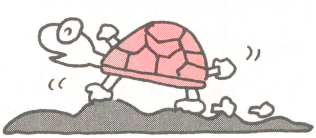
Aグループの
2番目のキャラクタを
3の配色番号で使う
ということだよ



こんな絵を作ってみたから、画面に出してためしてごらん。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0				G 6												
1	G 5											G 5				
2							G 5							G 6		
3				I 33						G 6						
4	G 6		I 03	I 13	I 23											
5			I 03	I 13	I 23		G 6									
6			I 03	I 13	I 23								G 00	G 20		
7		I 03	I 13	I 13	I 13	I 23				G 5		G 00	G 20	G 20		
8		I 03	I 13	I 13	I 13	I 23										
9	I 03	I 13	I 13	I 13	I 13	I 13	I 23									
10	I 03	I 13	I 13	I 13	I 13	I 13	I 23									
11	I 03	I 13	I 13	I 13	I 13	I 13	I 23									
12	I 13	I 13	I 13	I 13	I 13	I 13	I 13					G 01	G 00	G 10		
13	I 13	I 13	I 13	I 13	I 13	I 13	I 13					G 20	G 20	G 20		
14	I 13	I 13	I 13	I 13	I 13	I 13	I 13				G 00	G 20	G 20	G 20		
15			H 51	H 51	H 51				G 00	G 20	G 20	G 20	G 20	G 20		
16			H 51	H 51	H 51			G 00	G 20	G 20	G 20	G 20	G 20	G 20		

このつづきを作ってみないかい？
なに!?　ボクの方がうまいって？
じゃあ作ってごらん



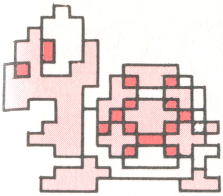
V 3でパワーアップだ！

◆便利な拡張ベーシック

行番号を順番にそろえたり 自動的に発生させてくれたり

メモリは2倍、内蔵ゲームも4つ入っているファミリーベーシックV 3。便利なベーシック命令が、たくさん使えるようになったんだよ。どんな命令があるか、かけ足で教えてあげよう。

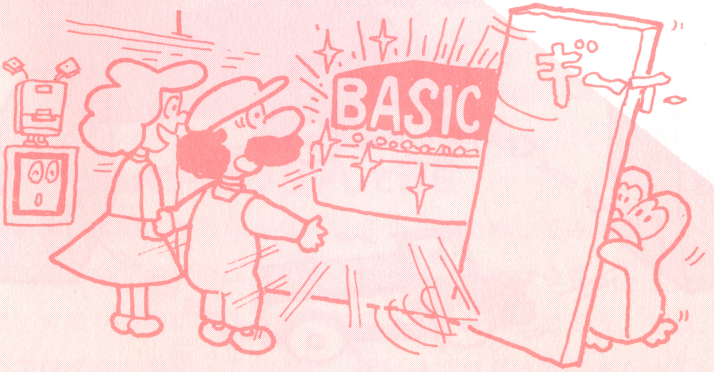
- AUTO（オート）命令……プログラムの行番号を自動的に発生させてくれる。
- RENUM（リナンバー）命令……プログラムの行番号を10、20、30……というように、10区切りで整理してくれる。
- DELETE（デリート）命令……いらないプログラムの行をまとめて消してくれる。DELETE 100～200なら、100行から200行までのプログラムをバッサリだ。
- CLICK ON／OFF（クリック オン／オフ）命令……CLICK ONなら、キー入力する音を出すし、それからCLICK OFFなら音を出さないというわけ。
- GAME（ゲーム）命令……ダイレクトモードで命令すれば、内蔵ゲームがすぐ楽しめるよ。
- FIND（ファインド）命令……FIND Aなら、プログラムの中の変数Aをさがし出して表示してくれる。

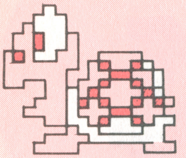


PART 4

ファミリーベーシックしよう

ファミコンにキーボードまで手に入れたきみは、もうただのゲームマニアじゃない。本格的なコンピュータワールドの扉を開いたのだ。奥は深いぞ。そして、その第一歩が、これから勉強しようというBASICということばなのだ。





4.1 行ってもどってまた行くくり返し



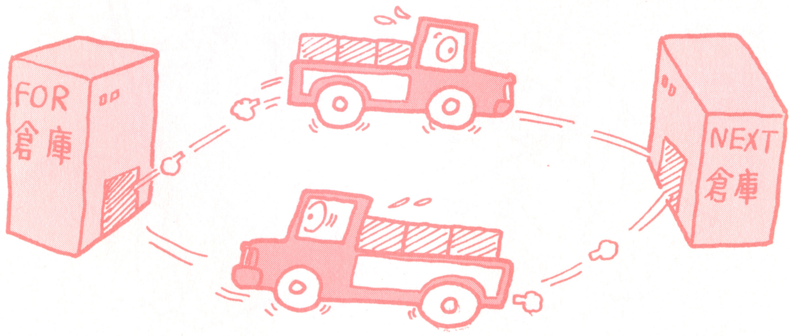
同じ仕事をなん度もなん度もくり返すたびにいちいち命令するのはめんどうだよ

■FOR〜NEXT（フォー〜ネクスト）

♥FORとNEXTにはさまれた仕事をくり返す

ベーシックにはたくさんの命令があるけれど、その中でも、もっとも基本的な命令というのが、いくつかある。まず、それらをマスターしないことには、プログラムは組めないぞ。

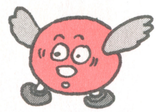
最初はFOR〜NEXTという命令だ。この命令は、FORとNEXTの間にはさまれた仕事を、指定した回数だけくり返す命令だ。



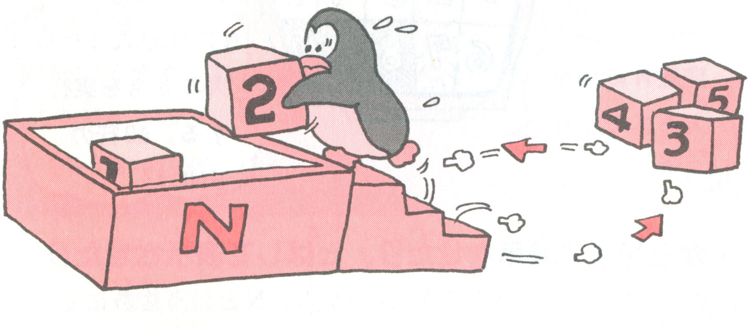
ファミコン運送会社は、FOR倉庫からNEXT倉庫まで、5回トラックで荷物を運ばなければならないとする。「荷物を運ぶ」という仕事を5回くり返すのだからこれをベーシックのFOR～NEXT命令であらわすと次のようになる。

```
FOR　N＝1　TO　5…………くり返される回数だよ
        荷物を運ぶという仕事
NEXT
```

（行って／もどって）



1回目、2回目……と数えなければならないね。このくり返しの数を数えてくれるのが変数Nだ。「1　TO　5」は、「変数Nに、1から5までの数が入るまで、仕事をくり返せ」ということなんだ。



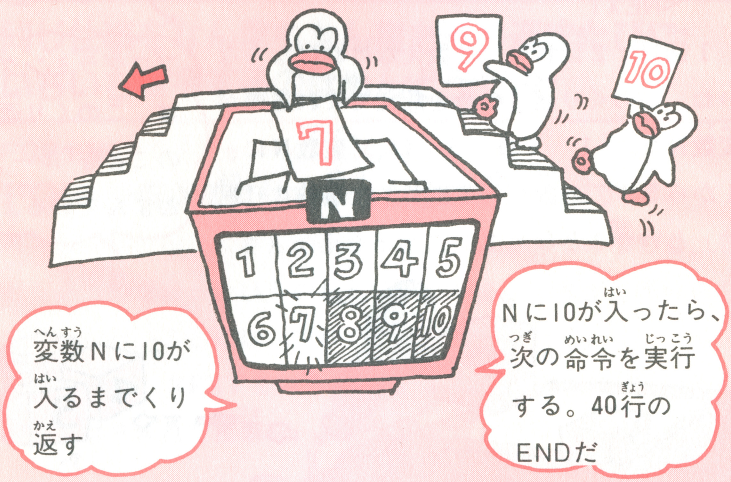
♥NEWでメモリをクリアしてから

では、実際にかんたんなプログラムでためしてみよう。その前に、新しいプログラムを入力するときは、かならず N E W ↵ で、メモリの中にある前のプログラムを消しておこう。もし、前のプログラムが残っていると、新しいプログラムとまざってしま

い、正しく実行されないことがあるからね。

消したくないプログラムは、カセットにセーブしておけばいいね。

```
10 FOR N=1 TO 10
20 PRINT N
30 NEXT
40 END
```



♥1から10まで表示させたり、とばして表示させたり

Nはもうわかるね。PRINT　Nだから、Nという変数に1から10まで順に送られてきた数を画面に表示させるんだ。Nに10が入ったら、ループは終わり。次の命令に移るよ。ここではEND命令が入っていて、プログラムの実行を終了させているよ。

■とびとびに表示させてみよう

こんな使い方もあるよ。上のプログラムで10行を、次のように

かえて、RUNしてごらん。

```
FOR  N=1 TO 10 STEP 2
```

1から10までの数が、1、3、5、7、9と1ずつとばして表示されたね。STEPは、ホップ・ステップ・ジャンプのステップ。2は増分といって、Nに2ずつたした数を表示しろということ。つまり、1の次は3、3の次は5というわけだ。9の次は、11だけど、「TO 10」の10の値より大きくなってしまうので、このループは、9を表示したところでおしまいだ。

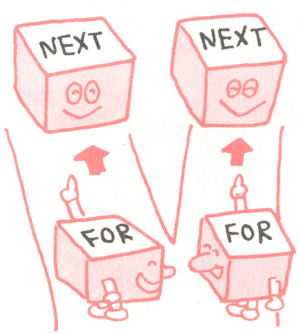


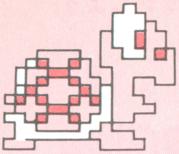
■FOR〜NEXTでしてはいけないこと

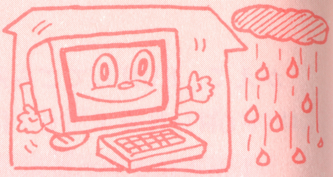
ループの中に、別のループを入れることもできるけど、ぜったいに交差してはいけないぞ。

```
FOR  N=1 TO 10
FOR  I=10 TO 20
  ∫
NEXT
NEXT
```

中に入れるFOR〜NEXT文は、完全に外側のNEXTの中に入っていなければだめだよ。







4.2 もし雨だったらファミコンしよう

> **もしこうだったらこうしよう**
> **もしああだったらああしようということだ**

■IF～THEN（イフ～ゼン）

♥仕事の内容も条件しだいでかわるよ

もし、あした晴れたら野球をしよう。雨だったらファミコンゲームだ。1,000円もっておつかいに行って、もし、おつりが100円以上あったらコーラを買おう。

「もし～だったら」というのは、きみたちのまわりにもたくさんあるね。この「もし～だったら」を条件というんだね。晴れたら、雨なら、おつりが100円あれば………みんな条件だ。そして、この条件によって、きみたちの行動だってちがってくるだろう。





■IF～THENとIF～THEN～ELSE

同じように、コンピュータに条件を与えて、その条件によって、いろいろな仕事をしてもらおうというのが、IF～THEN命令だ。

IF文には大きく分けて2とおりの文型がある。

①IF　条件　THEN　文または行番号

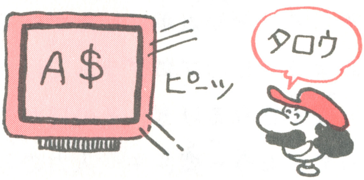
②IF　条件　THEN　文または行番号　ELES文または行番号

①は、「与えられた条件に合っていたら、THENのあとの文や行番号を実行しよう」という命令。

②は、途中までは①と同じだね。それに、「そうでなかったら、ELSEのあとを実行しよう」というのが加わっただけだよ。

では、実際にプログラムでどのように使うのかを説明しよう。

```
IF　A＝1　THEN　BEEP
```



このプログラムは、「もし、変数Aに1が入っていたら、音を鳴らしなさい」という意味だよ。

●BEEP（ビープ）命令

“ピー”という音を1回鳴らす命令。2回鳴らしたければ、

BEEP：BEEP　3回なら――――――





■キーボードから入力した数が条件に合っているかどうか

A＝1を文字変数に変えてもできるさ。

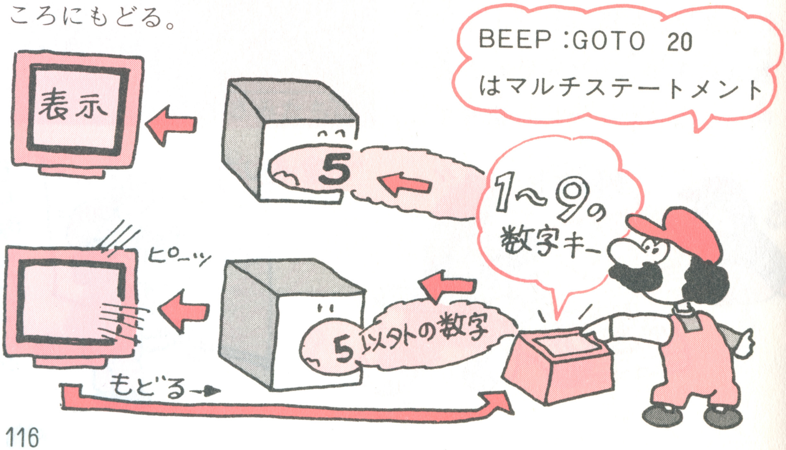
IF　A$＝〝タロウ〟　THEN　BEEP　GOTO　20

「もしA$がタロウなら音を出せ」ということだね。そうでなければ、GOTOのあとの20行を実行するよ。

もう少し具体的なショートプログラムでためしてみようか。

```
10  PRINT 〝1－9ノキーヲオシテクダサイ〟
20  Y$＝INKEY $（0）
30  IF  Y$<>〝5〟THEN  BEEP：GOTO  20
40  PRINT  〝5ガオサレマシタ〟
50  GOTO  10
```

入力したら、R U N ↵ だ。そして、画面表示にしたがって、1から9までの数字キーを押してみよう。このプログラムでは、5を押すと「5ガオサレマシタ」と表示されるけど、そのほかの数字キーを押すと、ピーと音が鳴って、もう一度、キー入力のところにもどる。



■IF文に使われる判断記号

ここでは、押された数が条件になるんだね。30行でY＄の中に“5”という文字が入ったかどうかを判断しているんだ。その判断する命令が＜＞という記号で、「等しくない」という意味だよ。

そのほか、こんな記号で判断されることが多いからおぼえておこう。

A＝B	AとBは等しい
A＜＞B	AとBは等しくない
A＞B	AはBより大きい
A＜A	AはBより小さい
A＞＝B	AはBより大きいか、または等しい（以上）
A＜＝B	AはBより小さいか、または等しい（以下）

演算記号も使えるよ。使い方は、

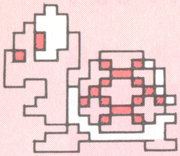
```
IF　A＝B＊2　THEN～
IF　(A＋B)＝8　THEN～
IF　A（I）＜50　THEN～
```

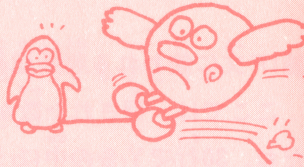
●ポイント

INKEY＄については、あとでくわしく説明するから、いまはキー入力するときの命令だとおぼえておけばいいよ。









4.3 とびとび実行のジャンプ命令

エイッとジャンプしてずっと先の行を実行と思ったらまたもどってきたりして

■GOTO（ゴーツー）

♥飛んだ先からどんどん先へ、行ったきりもどらない

プログラムは、行番号の小さいものから順に実行していく、ということは、もう知っているよね。ところが、順序を気にかけずに、好きな行番号のところへ行ってしまう場合がある。

それがGOTO命令だ。GOTO命令は、ジャンプの名選手。

```
10  GOTO  100
```

とすれば、10行のあとに20行があっても、30行があっても、見向



きもしないで、GOTOの次に書かれた100という行にジャンプして、その100行の仕事を実行するというわけだ。そのあとも、やはり20行にも30行にも知らんふりで、100行からあとのプログラムを実行する。でも、大きい行番号から小さい行番号にもジャンプできるから、小さい行番号にもどしたければ、またGOTOを使うよ。

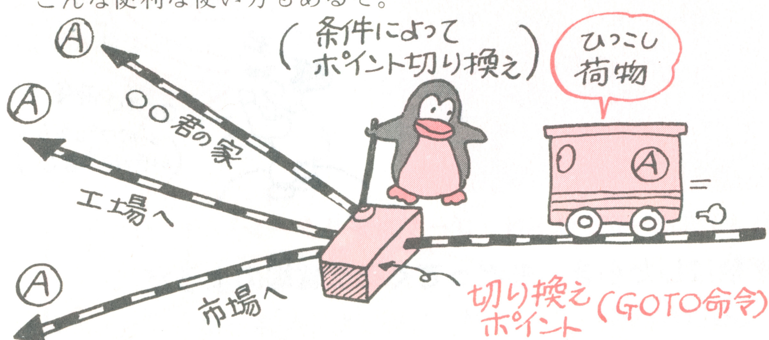
```
10  GOTO  30
20  PRINT   "ABC"
30  PRINT   "DFG"
40  END
```



RUNすると、20行をとばして、すぐに30行へジャンプ、40行でEND。ABCは永遠に書かれない。

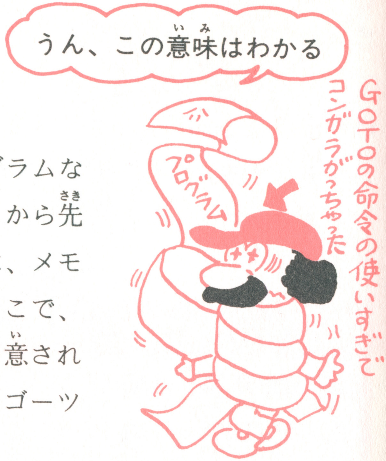
♥電車の切りかえポイントの役目もするよ

こんな便利な使い方もあるぞ。



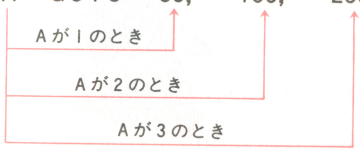
Aの貨車は変数だ。Aに入った数によって、飛び先をちがえたいときは？　そう、Aの条件によって変えればいいのだね。

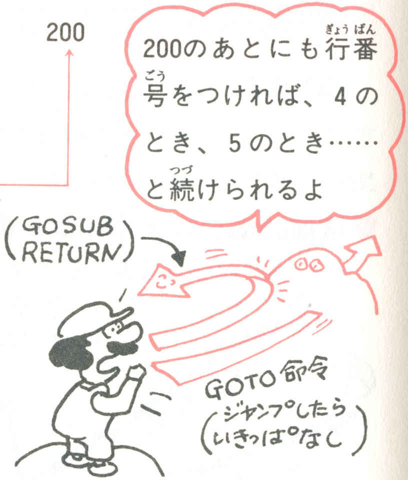
```
IF  A=1  THEN  50
IF  A=2  THEN  100
IF  A=3  THEN  200
```



でも、このぐらいの短いプログラムならいいけれど、切りかえポイントから先がいくつにも分かれているときは、メモリのむだ使いになってしまう。そこで、さらに便利な命令がちゃあんと用意されているのさ。ON～GOTO（オン～ゴーツー）という命令だ。

```
ON  A  GOTO  50,  100,  200
```





●ポイント

GOTO命令もあまり使いすぎると、プログラムがこんがらがっちゃうぞ。ほどほどにね。

■GOSUB～RETURN（ゴーサブ～リターン）

♥飛ばしたらまたもどってくる伝書鳩のようだ

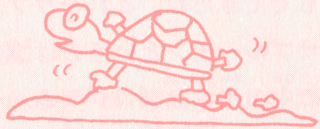
GOTO命令は、ジャンプして行ったら行きっぱなし。もとにもどってはこない。でも、ジャンプ命令の中には、行ったらまたもどってくるという、かしこい命令がある。それがGOSUB～RET

URN命令なのだ。

ちょっと長いプログラムだけど、よく見るとわかるよ。入力したら、さっそくRUNして、プログラムと見くらべてごらん。

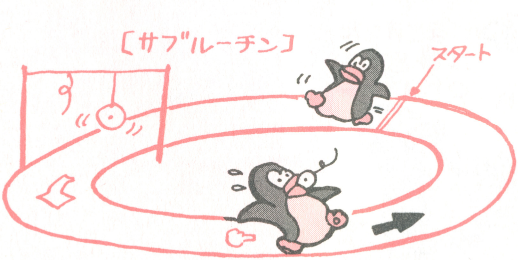
```
10  A=5:B=8
20  GOSUB 80
30  A=7:B=6
40  GOSUB 80
50  A=4:B=9
60  GOSUB 80
70  END
80  C=A*B
90  PRINT A;"*";B;"=";C
100 RETURN
```

```
RUN
5 * 8 =40
7 * 6 =42
4 * 9 =36
```



アレ!? へんなところにENDがあるなあ。では、プログラムがどのように実行されて、画面表示のようになったか見てみよう。実行の順は次のようになっているんだよ。

10・20→ 80〜100 →30・40→ 80〜100 →50・60→ 80〜100 →70



赤い矢印は、
GOSUB 80
黒の矢印は
RETURN

♥なんどもさせる仕事はサブルーチンに

GOSUB命令もジャンプ命令だから、「GOSUB　80」で、80行からの命令を実行するところまでは同じ。ところが、100行でRETURNという命令が出てくると、そのGOSUB命令のあとに続く命令にもどって、それを実行するんだ。

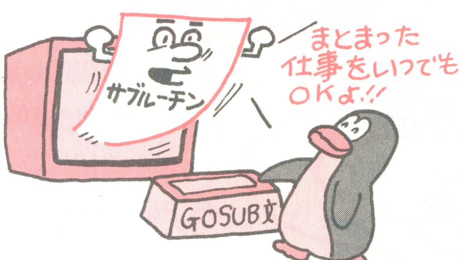
20行のGOSUB命令で、80行から100行を実行したあと、こんどは30行の命令を実行するというわけさ。

ところで、実行順序の図を見ると、 80～100 というのが、3回もでてきたね。これをサブルーチンというんだ。

サブルーチンというのは、プログラムの中の小さなプログラム。それだけでちゃんと一人前の仕事をもっている。

変数に代入する数さえかえれば、このプログラムのサブルーチンは、いつでも同じ計算をしてくれる。

プログラムの中で、なん度も使うような内容があったら、サブルーチンにしておこう。いつでもGOSUB文で呼び出すことができて、とても便利だ。

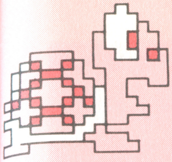


●ポイント

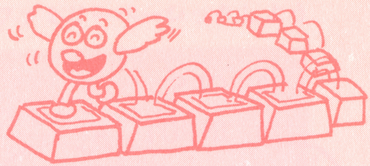
GOSUBは、GO（行け）SUB（サブルーチンへ）とおぼえておけばいいね。

●ポイント

サブルーチンを使うのに、GOTOで飛ばしたり、RUNで実行させることはできないよ。かならずGOSUB、そしてサブルーチンのあとにはRETURNを忘れずにね。



4.4 キーからデータ入力



実行中に"データを入れてください"
キーボードからインプットすればいい

■INPUT（インプット）

♥？マークが出たらデータ入力だ

プログラム実行中に、キーボードからデータを入力して、コンピュータに仕事をさせる命令が、いくつかある。そのはじめは、INPUT命令だ。

プログラムを実行中に、このINPUT命令のところにくると、画面はストップし、？マークがでる。「さあ、データを入力してください」という合図だ。

INPUT命令のあとには、かならず変数が置かれている。この変数に、インプットされたデータが入るんだよ。



```
10  CLS
20  INPUT  A ─────変数Aに数を代入せよ
30  INPUT  B ─────変数Bに数を代入せよ
40  PRINT  A＊B ───計算をして画面に表示せよ
```

RUNさせると、画面は“？”（クエスションマーク）を出してとまったままだね。このままなにもしなければ、永遠にとまったままだ。

そこで、1でも2でもいいから、キーボードの数字キーを押して、きみの好きな数字を入力しよう。この数字が変数Aに代入される。

また、？が出たね。もうひとつ数字をキー入力しよう。この数字が変数Bに代入される。

●ポイント

キーボードからデータを入力しおわったら、かならず(RETURN)キーで、終了の合図を出してやることを忘れずに。これを忘れると、永遠にキー入力待ちだ。

それから40行の仕事、A×Bのかけ算をして、その答えを画面に表示してくれる。



●メッセージつきINPUT命令

同じ仕事でも、もう少し親切なプログラムにしてあげようか。

```
10  CLS
20  INPUT   "A＝"；A
30  INPUT   "B＝"；B
40  PRINT    A＊B
```



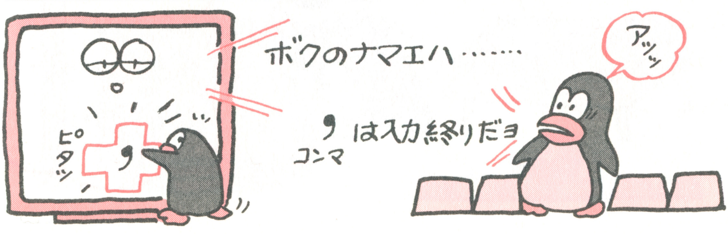
これをメッセージつきのＩＮＰＵＴとおぼえておこう。このときは、変数の前に「；」(セミコロン)をつけ忘れないでね。メッセージは「〝 〃」(ダブルクォーテーション）で囲んでつけておく。

●文字変数にデータを入れるときは「，」(コンマ)に注意だ

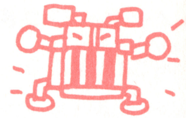
もちろん、文字変数だったら、文字データのキー入力だ。ただし、文字データを代入するときは、「，」(コンマ）は代入できない。「，」は、「ここでキー入力は終り」という合図の役目をしてしまう。

```
10  CLS
20  INPUT "A＋B＝" ；A，B
30  PRINT  A＋B
```

これは数の代入だから、「5，8」とコンマで区切って代入する。



```
10 CLS
20 INPUT "A$";A$
30 PRINT A$
```



文字の代入だが、「ABC，DEF」と代入しても、コンマがあるため、ABCだけで代入終りと判断されてしまうよ。

■LINPUT（ラインインプット）

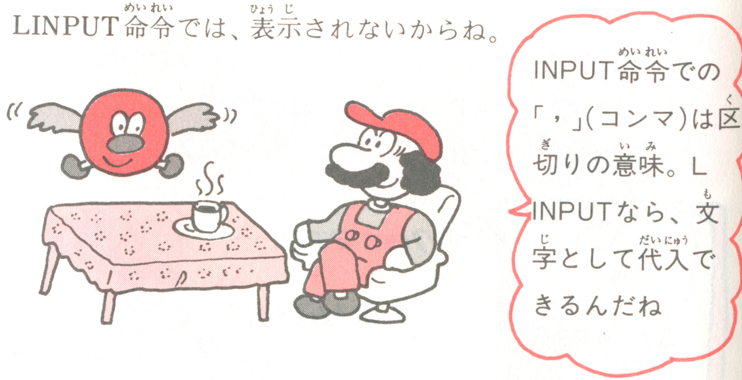
♥メッセージごと代入できるし「，」も代入できる

文字の代入のときに、「，」(コンマ)が代入できないのは、どうも不便だ。だって、実際には、文字とコンマを使ったデータも多いからね。でも、同じINPUTでも、LINPUT命令を使えば、文字といっしょに、コンマを代入できるよ。これが大きな特徴だ。

そのほかにも、INPUT命令と違う点がいくつかあるよ。INPUT命令のときは、メッセージはあらかじめ別に入れておいて、データだけをキー入力したけど、LINPUT命令なら、あとでキーボードから代入するときに、メッセージまで代入できるんだ。

代入できる文字数は、メッセージをふくめて31文字以内だ。

もうひとつ、INPUT命令のときは、？マークが表示されたけど、LINPUT命令では、表示されないからね。



■INKEY＄（インキードル）

♥1文字だけを受けつけて実行する

INPUT命令は、いくつもの文字や数字をデータとして入力することができたね。

INKEY＄という命令は、キーボードから入力することは同じだけど、たった1文字の代入しか受けつけてくれない。

キー入力する文字や数字も、INPUT命令がデータだったのに対して、INKEY＄の1文字は命令といったほうが適当かもしれないな。

```
10 CLS
20 PRINT 〝サイゲームヲシマスカ（Y／N）〟
30 A＄＝INKEY＄
40 IF A＄＝〝Y〟 THEN 70
50 IF A＄＝〝N〟 THEN 80
60 GOTO 30
70 PRINT 〝イエス（Y）〟：END
80 PRINT 〝ノー （N）〟：END
```



● RETURN なくてもすぐ実行

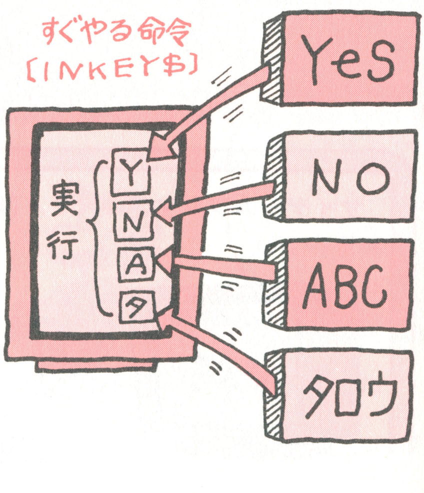
きみがYかNを入力するまで、ずっと待っているはずだが、INPUT命令とちがって、そこでずっと待っているわけじゃない。30行で、きみがなにも入力しなければ、入力されなかったものとして、実は通りすぎてしまうのだ。

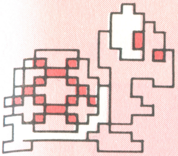
> **●ポイント**
>
> INKEY＄命令の書きかたは、
>
> **文字変数＝INKEY＄**

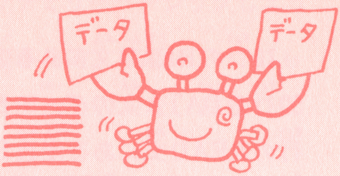
それではたいへんだから、GOTO命令で30行にもどしてやるんだ。YかNが入力されたら、あとはわかるね。40行、50行のIF～THEN命令のとおりだよ。

このINKEY＄には、？マークもでないし、メッセージもつけられないかわりに、RETURN キーを押さなくても、入力と同時にすばやく実行してくれる。なんのためにキー入力するのかは、INKEY＄命令の前にあらかじめPRINT文で打ちこんでおくようにすればいいよ。

> **●ポイント**
>
> ループを作らずに、1文字入力するまで通りすぎずに待っていてほしいときは、
>
> **A＄＝INKEY＄(0)**
>
> とすると、キー入力があるまで待ちつづけるよ。







4.5 並んだデータを順番どおりに呼び出す

たくさんのデータをインプットしておけば あとから必要なデータを取り出せる

■READ～DATA（リード～データ）

♥変数の順とデータの順は一致する

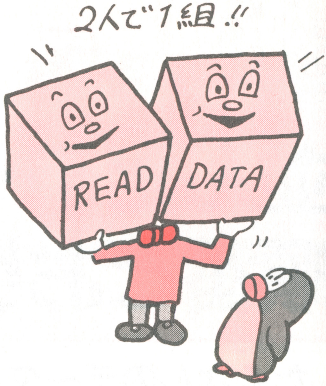
READ～DATA命令は、DATA文に書かれたたくさんのデータを呼びだして、READ文にある変数に順番どおりに代入する命令だ。かんたんなプログラムで説明しよう。

```
10  READ   A
20  READ   B
30  PRINT   "A="   ; A
40  PRINT   "B="   ; B
50  DATA   8
60  DATA   20
```

READは"読め"という英語。"データを読め"という命令だ



10行のREAD文にある変数Aに、50行のDATA文のデータ8が、20行のREAD文にある変数Bに、60行のデータ20が代入される。つまり、READ文の最初の変数は、DATA文の最初のデータを読み込むということだ。変数の順とデータの順は一致するということがわかるね。



♥ファミコン電話帳プログラム

ファミコンを電話帳がわりに使うプログラムを考えてみたよ。

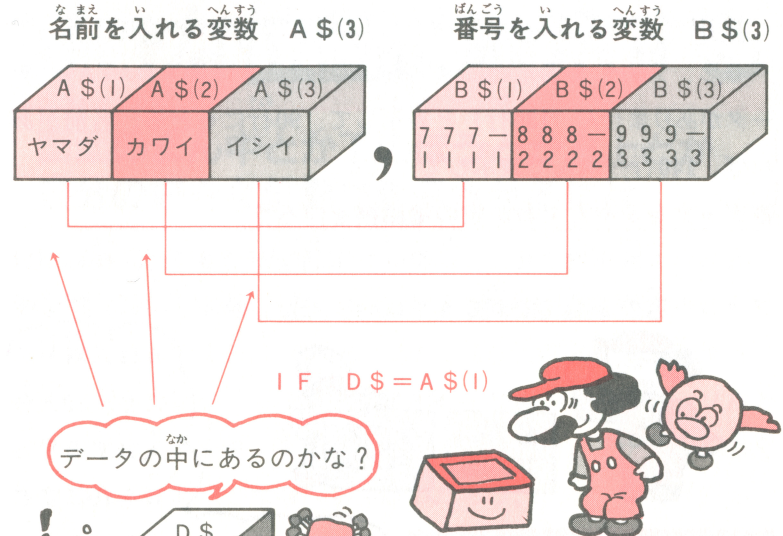
```
10  CLS
20  DIM  A$(3), B$(3)
30  FOR  N=1  TO  3
40  READ  A$(N), B$(N)
50  NEXT
60  DATA    ヤマダ, 777−1111
70  DATA    カワイ, 888−2222
80  DATA    イシイ, 999−3333
90  INPUT  "ナマエハ" ; D$
100 FOR  I=1  TO  3
110 IF  D$=A$(I)  THEN  Y=I :GOTO  140
120 NEXT
130 PRINT  "NO DATA" :GOTO  90
140 PRINT  A$(I), B$(I)
150 PRINT  "モウイチド  シマスカ (Y/N)"
160 E$=INKEY$(0)
170 IF  E$="Y" THEN  90
180 END
```

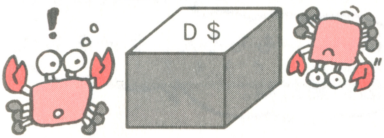
30〜40行: FOR〜NEXTで3つのデータを読み込む

60〜80行: 名前はA$に、番号はB$に、Nが3になるまで、読み込まれるよ

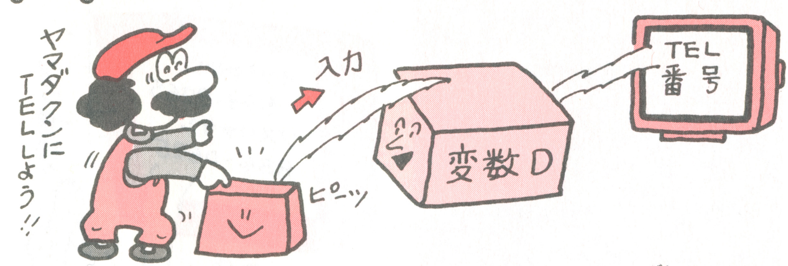
130行: データはありませんという表示だ

さあ、見たことのある命令ばかり出てきたぞ。変数がいくつもでてくるから、それぞれの変数にはなにが入るのかを、注意してプログラムを読んでほしい。





キーボードから調べたい人の名前を入力すると、変数D$に入るよ。



A$（3）、B$（3）だから、A$（1）には、1番目のDATA文の名前が、B$（1）には、同じように1番目のDATA文の電話番号が入るね。

Ｄ＄に、INPUT命令でキーボードから名前を入力すると、110行で「ＩＦ　Ｄ＄＝Ａ＄（Ｉ）」として、入力した名前がＡ＄の中にあるかどうかさがす。110行で、「１番目から３番目までさがせ」とFOR～NEXTのループで命令していることがわかるね。

もし、入力した名前と同じ名前がなければ、「NO DATA」（データがありません。130行）と表示して、90行にもどる。名前があれば、140行へ飛んで表示してくれる。

■データをふやしてわが家の電話帳を作ろう

もっと人数を増やせば、本格的な電話帳ができるぞ。今のプログラムを次のように手直しすればいいんだ。

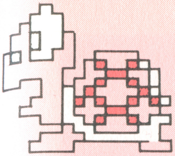


●ポイント

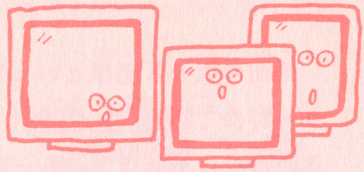
１．変数とデータの数はかならず同じであること。

２．文字変数で呼び出すのは文字データ、数値変数で呼び出すのは数のデータ。変数とデータの種類をまちがえるな。変数「Ａ」に、データ「タロウ」なんて、大エラーだぞ。





4.6 好きな位置に表示する

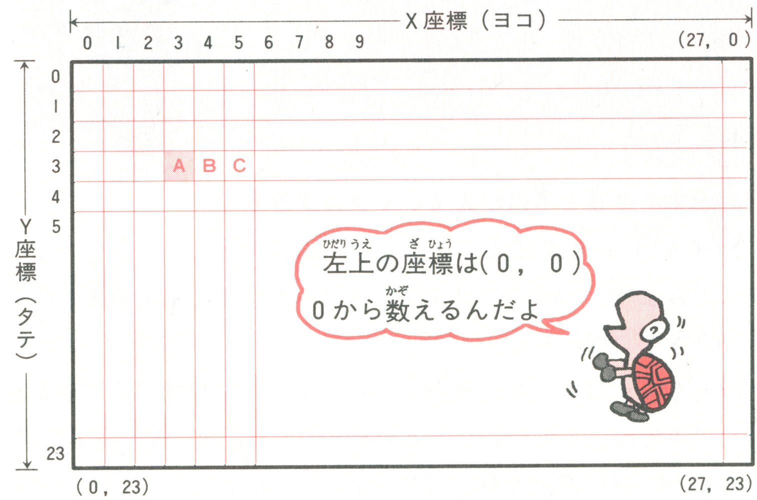


画面のまん中とか下のほうとか右上とか自分の好きな位置に表示させるには

■LOCATE（ロケート）

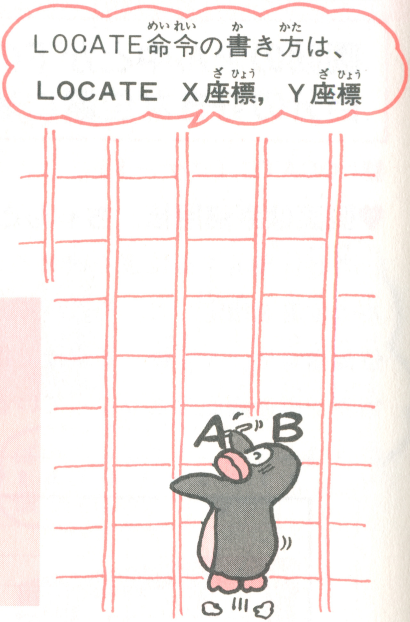
♥画面は原稿用紙、ちゃんとマス目があるんだよ

画面に表示する位置を決めるのが、LOCATE命令だ。カーソルの位置を指定して、そこから表示をはじめるんだ。カーソルの位置指定は、X座標、Y座標で指定する。では、ファミコンの文字表示用の画面はどうなっているのだろう。



画面の横をX座標といって、0～27まで28個に区切られている。たてはY座標で、0～23までの24個だ。このX座標とY座標が交わって、画面には合計28×24の672個のマス目があるわけだ。この範囲の中で、表示する位置を決めるんだよ。

```
10  CLS
20  LOCATE  3, 3
30  PRINT  "ABC"
40  END
```



INPUT文でキー入力した文字や数の位置だって、自由にかえられるよ。

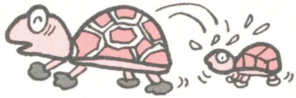
```
10  CLS
20  LOCATE  2, 3
30  INPUT  "A=" ; A
40  LOCATE  4, 5
50  PRINT  A
60  END
```

■表示する位置を順にかえていくプログラム

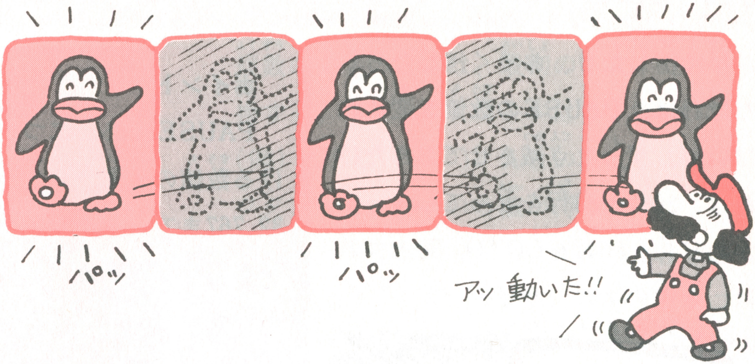
LOCATE命令を使って、少しゲームに使えるようないたずらをしてみよう。"＊"(アスタリスクマーク)を、画面の左から右へ動かすプログラムだ。

みんなもうおなじみの命令ばかりだから、自分でプログラムを読んでみてね。このプログラムのポイントは、20～30行と60～70行のFOR～NEXTとLOCATEだよ。

```
10  CLS
20  FOR  N=0  TO  26
30  LOCATE  N, 5
40  PRINT  "*"
50  BEEP:PAUSE  10
60  LOCATE  N, 5
70  PRINT  " "
80  NEXT
90  LOCATE  N, 5
100 PRINT  " "
110 END
```



70行の“　”の中の空白はちゃんと意味があるんだね。＊が左から右へ動くように見せるのは、書いては消して書いては消してのくり返しだ。空白を表示すれば消したことになるね。こんな使い方もおぼえておこう。



50行に、新しい命令がひとつ入っているね。PAUSE（ポーズ）命令は、プログラムの実行を一時中断してくれるんだ。写真を撮るときの、「ハイ、ポーズ！」と同じだよ。10という数値は、中断している時間の長さ。秒や分のことじゃないよ。コンピュータ専用の時間だ。ためしに、10を100にかえてみるとわかるよ。

■ネオンサインのプログラム

もうひとつ、ネオンサインふうのプログラムを考えてみた。ABCDEFのかわりに、きみの名前でも入れてごらん。

```
10  CLS
20  FOR  N＝1  TO  10
30  LOCATE  10, 10
40  PRINT  "ABCDEF"
50  BEEP：PAUSE  50
60  LOCATE  10, 10
70  PRINT  "        "
80  NEXT
90  END
```

（20行について）フラッシュさせる回数。このFOR～NEXTの使いかたをカウンターループなどというよ

（70行について）空白表示の活用だ。40行の文字数分と同じかそれより長めに入力しよう

早く消したり、おそく消したりしたければ、PAUSEの数値を変えればいい。ずっと点滅させておきたければ、20行と80行をとってしまって、かわりに、80行をGOTO　30とすれば、永遠のネオンサインだ。



●ポイント

PAUSEのあとにつける数値の範囲は0～32,767までだ。また数値を省略したときは、なにかキー入力をしないかぎり、休止したままだ。そのときは、どれでもいいからキーを押してごらん。次の行へ進むよ。

関数はプログラミングのための高度なテクニックだ。上達するにしたがってたくさん使うようになるが、はじめはまる暗記でいこう。

●数値関数

ABS(x)……………数式の絶対値を得る。
（例）ABS（－2）　値は2。

SGN(x)……………Xの値がX＞0のときに1、X＝0のときに0、X＜0のときに－1を関数の値とする。

RND(x)……………乱数を発生させる。
（例）RND（5）　0から4までの乱数を発生させる。

●文字関数

ASC(文字列)………文字をキャラクタコードに直す。
複数の文字のときは、最初の1文字だけを変換する。キャラクタコードは0～255まで。
（例）ASC（"A"）　Aという文字のキャラクタコード、65という値が得られる。

CHR$(x)…………キャラクタコードから文字を引き出す。
（例）A$＝CHR$（65）　キャラクタコード65（10進コード）に相当する文字"A"をA$に代入する。xの範囲は32～255まで。

VAL(文字列)………文字として扱われている数字を、数値として扱えるように直す。
（例）VAL（"273"）　文字"273"を数値273に直す。

STR$(x)…………VALの逆。数値を文字列に直す。

(例) A$=STR$(8)　8は数ではなく、文字として扱われる。

HEX(x)……………xの数値を文字列に変換して、その結果が関数の値となる。xの値は−32768〜+32767まで。

(例) HEX$(13) 値は16進数の「D」。

LEFT$(文字列, n)…………文字列の左からn個の文字を取り出す。nは0〜31まで。

(例) LEFT$(BASIC, 2)　BAの2文字が関数の値となる。

RIGHT$(文字列, n)………文字列の右からn個の文字を取り出す。nは0〜31まで。

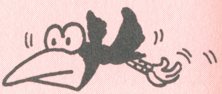
(例) RIGHT$(BASIC, 2) ICの2文字が関数の値となる。

MID$(文字列, n, n^1)………文字列の左からn番目の文字からはじまってn^1個の文字を取り出す。

(例) MID$(BASIC, 2, 2)
ASの2文字が関数の値となる。

LEN(文字列)………文字列の文字数をかぞえる。

(例) LEN(〝ABCD〟)　文字列〝ABCD〟の文字数をかぞえて関数の値とする。値は4。

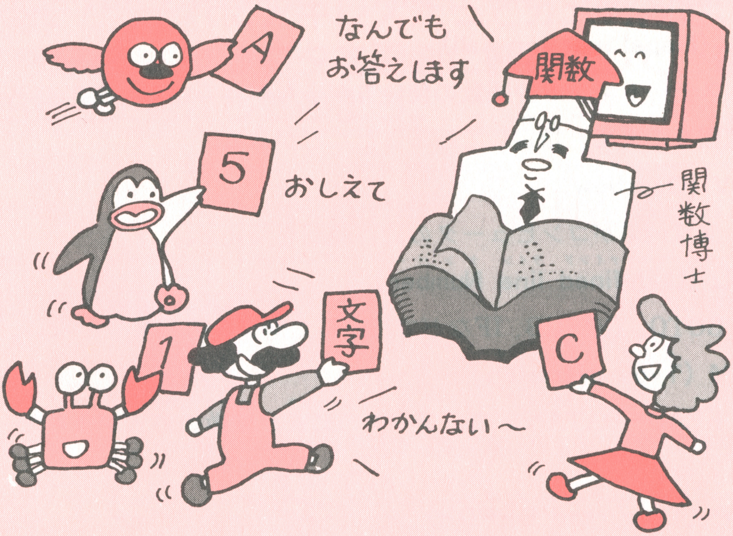


特殊関数

PEEK(アドレス)………指定したアドレスからデータを取り出す。読み出すデータは1バイト分（0～255の値)。

POS(数式)…………画面上の現在のカーソルの水平位置を調べる。範囲は0～27。

FRE…………………メモリの未使用領域の大きさを調べる。プログラムを変更したり加えたりすることで、値が変化する。



プログラムを実行中に、アレッ？突然ストップして、画面に変な表示があらわれた。これは、きみの作ったプログラムにどこかおかしいところがあるからだよ。でもだいじょうぶ。まちがいをおかしてもちゃんと画面に表示して教えてくれる。

NF（NEXT without FOR）

FORがないのにNEXTがある。

SN（Syntax error）

プログラムが文法に合っていないというエラー。実は、キー入力するときの打ちまちがいが多い。

RG（RETURN without GOSUB）

GOSUBがないのにRETURNがある。

OD（Out of DATA）

READで読むべきデータがDATA文に用意されていない。

IL（Illegal function call）

ステートメントや関数の使い方がまちがっている、指定できる範囲をこえて、指定してしまったなどがある。

OV（Overflow）

演算結果がコンピュータの許容範囲を超えた。

UL（Undefined line Number）

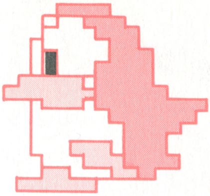
GOTO、GOSUB、IFなどで指定した飛び先の行番号がない。

TM（Type mismatch）

READ～DATAなどで、データの形式と変数の形式がまちがっている。

TP（Tape read ERROR）

カセットテープからデータが正しく読めない。

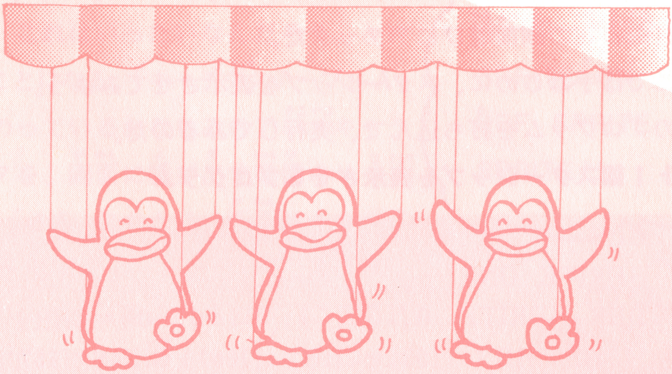


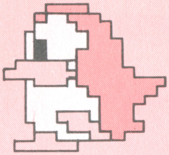
PART 5

キャラクタを自由に動かす

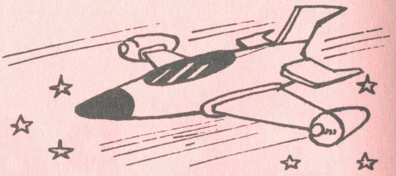
たのしいキャラクタいっぱいのファミリーベーシック。マリオもレディもペンペンもいるぞ！　それにアキレスだってカメだって。

好きなキャラクタを画面に呼び出して、自由自在に動かしてみよう。





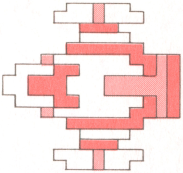
5.1 出てきたぞスターシップ



マリオもレディもアキレスもいるぞ！
かわいいペンペンとも遊べるのだ

♠まず最初にスターシップを表示させてみる

テレビの画面にいろいろなキャラクタを呼び出して、画面いっぱい、自由に跳びはねさせてみよう。「キャラクタテーブルA」にのっているどのキャラクタでもいい、きみが画面に表示させたいキャラクタを選んだら、キャラクタ呼び出し自由自在の画面遊びにアタックだ。



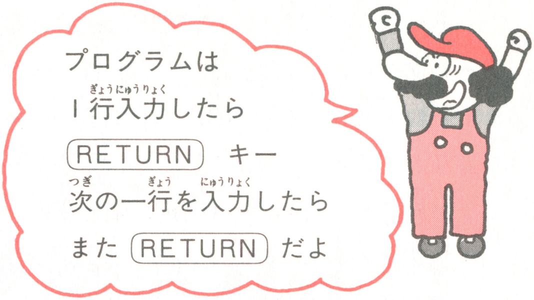
スターシップ

ここでは手はじめに、スターシップを表示させてみよう。次のプログラムを打ち込んで、実行してみるのだ。

リスト1 ■スターシップを表示させるプログラム

```
10  CLS
20  SPRITE ON
30  DEF SPRITE 0,(0,1,0,0,0)=CHR$
    (164)+CHR$ (165)+CHR$ (166)+CHR$ (167)
40  SPRITE 0,100,100
```

さて、どんなことになったかな？　ほら、ほら画面には、スターシップがおどり出て表示されたね。



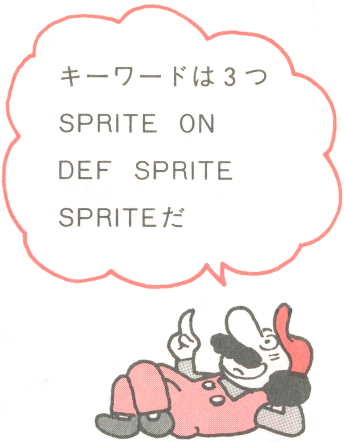
♠キーワードは3とおりの"SPRITE"

またあとでくわしく教えるけど、ここで、ごくかんたんに、プログラムを説明しておこう。

もう知っているはずだけど、10行のCLSは、画面をきれいにそうじしたのだったね。

そして、20行のSPRITE　ON、30行のDEF　SPRITE、40行のSPRITE、みんな"SPRITE"という文字がついているね。どうもここらあたりが、キャラクタを表示させるキーワードらしい。

最初に出てきたのが、SPRITE ONだ。これはいったいなんだろう、ということになる。「キャラクタテーブルA」にのっているおなじみのアニメキャラクタを、画面に呼び出す準備のサインだ。「さあ、これから、画面にキャラクタを出すぞ」、と宣言して、スプライトのスイッチを入れるのが、このSPRITE　ONだと考えておいていいだろう。

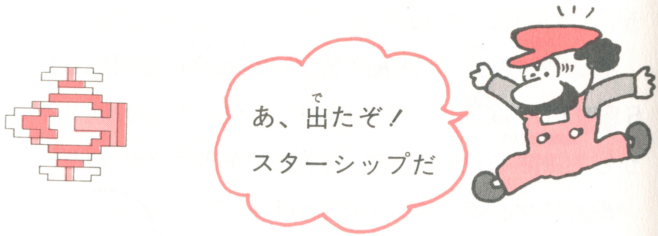


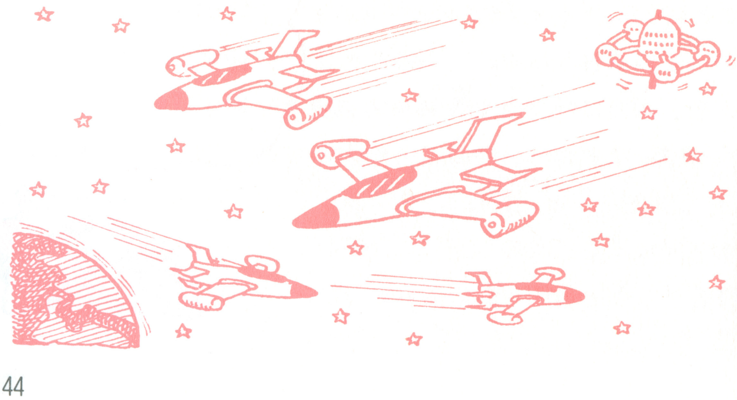
その次が、30行にあるDEF SPRITE。そのあとに、なんだかんだと数字が出てくるね。ここでは、DEF SPRITEの次は0。そのあとにも、たくさん数字や文字が出てくるけれど、DEF SPRITE 0なら、「このスプライトの番号は0だよ」という意味だとおぼえておこう。

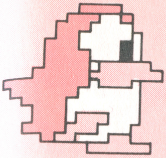
そして最後が、ただのSPRITEだが、これはアニメキャラクタを、指定した位置に表示しなさい、という意味だ。

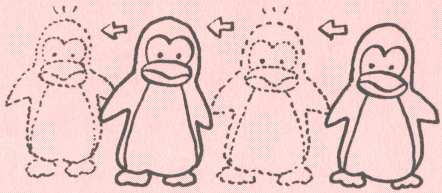
●ポイント

アニメキャラクタを画面に表示させるキーワードは“SPRITE ON”、“DEF SPRITE”と“SPRITEだ”。









5.2 スプライトのスイッチを入れよう

スプライト画面のあかりをつけたり消したり
キャラクタも表示されたり消えたりだ

■SPRITE ON（スプライト オン）

♠描いたアニメキャラクタが見えるようにする

　さっきのプログラムについて、もう少しくわしく説明しよう。SPRITE（スプライト）命令は、「キャラクタテーブルA」にあるいろいろなアニメキャラクタを、画面に表示させる命令だが、もっと正確にいえば、「スプライト画面」というキャラクタを表示させる画面があって、そこに表示させているんだ。

　スプライト画面に、きみの好きなアニメキャラクタを描いておいたとして考えてみよう。このスプライト画面は、あかりがついたり消えたりするようになっていて、画面のあかりがつけば、描いておいたキャラクタが表示されるし、あかりが消えていれば、そこに描かれているキャラクタも見えなくなってしまう、という寸法だ。

　スプライト画面にスイッチを入れて、あかりをつける命令が、SPRITE ONだ。こうしてあかりをつけておかなければ、スプラ

イト画面にキャラクタが描かれても、目には見えないということ。
だから、アニメキャラクタを画面に表示させるためには、かならずSPRITE ONであかりを入れておかなければならないのだよ。

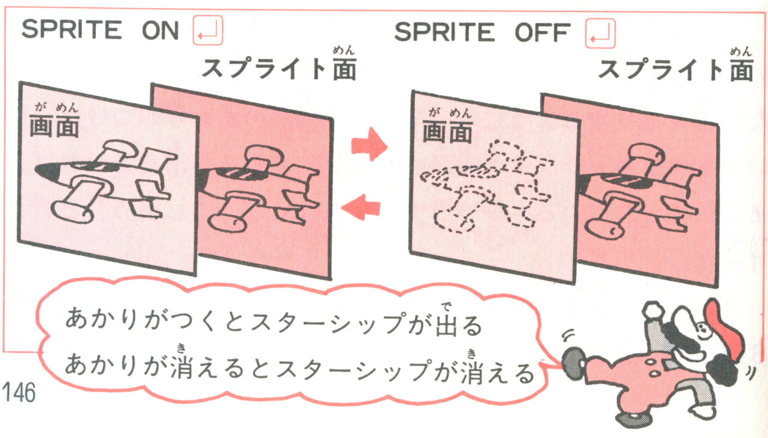
■SPRITE OFF（スプライト　オフ）

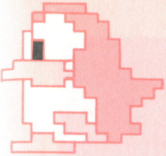
♠キャラクタは描かれたまま。ただ見えないだけ

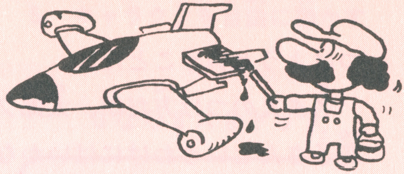
スプライト画面のあかりのスイッチを入れるのが、SPRITE ONだとしたら、入れたあかりのスイッチを切るのがSPRITE OFFだ。
スイッチを切ったのだから、スプライト画面に描かれたアニメキャラクタは、見えなくなってしまったけど、スプライト画面にはまだキャラクタは残っている。だから、もう一度SPRITE ONしてスイッチを入れれば、キャラクタはまた見えるようになるというしくみだよ。

●ポイント

SPRITE ON とSPRITE OFFはスプライト画面のあかりをつけたり消したりする。
あかりをつければ、キャラクタが見え、あかりを消せば、キャラクタは見えなくなるだけだ。







5.3 スターシップの大きさや色づけしよう

どんなキャラクタを どんなふうに表示させるかを決める

■DEF　SPRITE（デフ　スプライト）

♠まずスプライト番号で名まえをつけてやる

SPRITE　ONで、スプライト画面のあかりを入れたら、次はどんなアニメキャラクタを表示させるかだ。そのキャラクタに名まえをつけてあげたり、色をつけたり、大きさを決めてあげたりしよう。それが、30行のDEF　SPRITEだ。

```
30　DEF　SPRITE　0，（0，1，0，0，0）＝
    CHR＄（164)/+CHR＄（165)＋CHR＄（166)＋
    CHR＄（167)
```

やけに長くてむずかしそうだけど、意味がわかってしまえばかんたんだ。すぐにおぼえてしまうだろう。

DEF　SPRITE　0
は、表示させるキャラクタを「スプライト番号0」としようということだ

まず、DEF　SPRITE　0

これから表示させようというキャラクタの名まえを「SPRITE　0」と



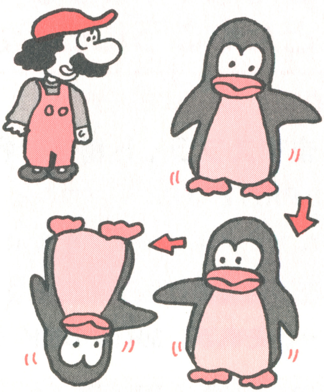
つけようということだ。

名まえのつけ方だが、太郎クン、花子さんというわけにはいかない。つけていい名まえは8種類、0～7までの1けたの数字だ。これがスプライト番号といわれるものだよ。

●ポイント
DEF SPRITEでつけるキャラクタの名まえは、0～7の数字で、これをスプライト番号というんだよ。

♠左右や上下の向きを逆にしたりもできる

名まえをつけたらその次は、表示させるキャラクタの色や大きさを決めてあげよう。表示させる順位やキャラクタをひっくり返したり裏返したりもできるよ。



DEF SPRITE　0の次を見てみよう。

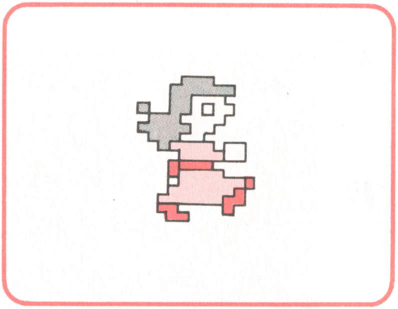
（0，1，0，0，0）と書いてあるね。この5個の数字の意味を左から順に書いておこう。

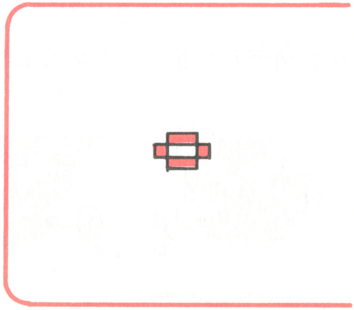
1番目の数字……キャラクタを表示させる色を決める。カラーチャートの配色番号0、1、2、3のうちから指定できるけど、キャラクタテーブルAにあるキャラクタと同じ配色なら0を指定すればいい。

2番目の数字……表示させる大きさを決める。0なら8×8ドット（1文字ぶんの大きさ）、1なら16×16ドット（4文字ぶんの大きさ）だ。

（0，**1**，0，0，0）

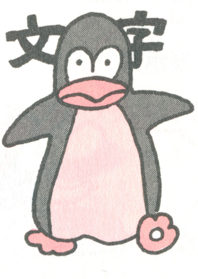
1なら16×16ドット　　　　0なら8×8ドット

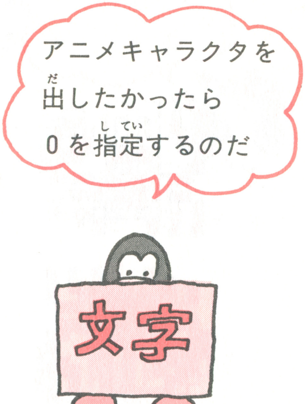






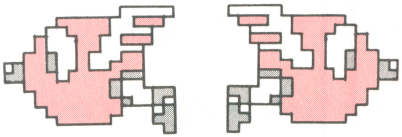
3番目の数字……文字などが表示されているバックグラウンド面と、アニメキャラクタが表示されるスプライト面が重なったとき、0ならキャラクタが前面に表示され、1なら文字が前面に表示されて、キャラクタは文字のかげにかくれるよ。



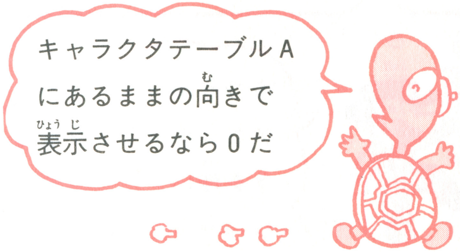


4番目の数字……キャラクタテーブルAにあるアニメキャラクタの左右を逆にしたかったら1、そのままの向き（左向き）で表示するなら0だ。

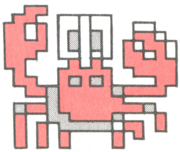
0を指示した表示　**1を指示した表示**



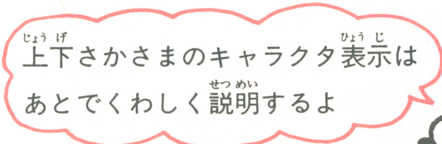
> **●ポイント**
>
> ただし、1にしたからといって、自動的に向きが反対のキャラクタが表示されるわけではない。
>
> 1にしたときは、それなりにほかの作業が必要だが、それはあとでくわしく説明しよう。

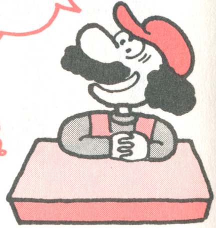


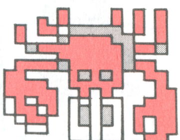
5番目の数字……キャラクタテーブルAのアニメキャラクタを上下逆に表示させるなら1、そのまま表示させるなら0だ。



0を指示した表示







1を指示した表示



■キャラクタを左右逆に表示

ここで、アニメキャラクタの左右を逆に表示させてみよう。たとえば、左向きのマリオにしてしまうのだ。

DEF SPRITE 0,（0, 1, 0, 1, 0）として、左右の逆向きを指定したあとだ。

左向きマリオのキャラクタ番号の指定は、左上、右上、左下、右下の順に0、1、2、3だから、

CHR＄（0）＋CHR＄（1）＋CHR＄（2）＋CHR＄（3）

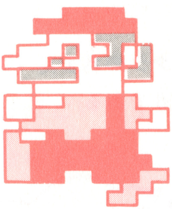
と、こうなるね。

これを右側のキャラクタと左側のキャラクタを入れ替えるのだから、指定の順序は、

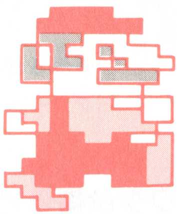
CHR＄（1）＋CHR＄（0）＋CHR＄（3）＋CHR＄（2）

と、こうすればいいわけだ。

左向きのマリオ



右向きのマリオ



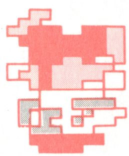


■キャラクタを上下逆に表示

キャラクタを上下逆に表示させるのも、左右逆に表示させるのと同じ方法だ。

マリオのキャラクタを上下逆に表示させるなら、

上下逆のマリオ

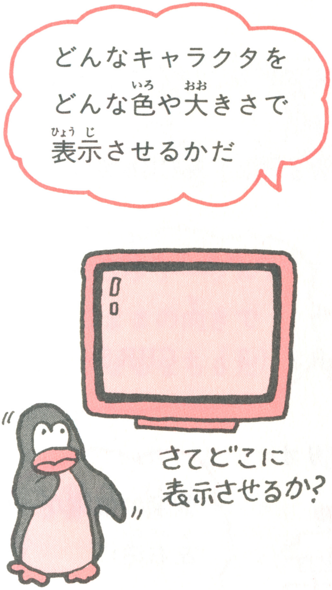


CHR＄（3）＋CHR＄（2）＋CHR＄（1）＋CHR＄（0）となるけれど、この理由はわかるね。

少し複雑になってきたから、ここでちょっとまとめてみよう。

```
30 DEF SPRITE 0,(0 1,0,0,0)
```

- 0（第1の0）：表示させるキャラクタの名まえ（スプライト番号）は0である
- 0（第2の0）：キャラクタテーブルと同じ配色なら0
- 1：ふつうのアニメキャラクタなら1。レーザーなら0
- 0（第4の0）：スプライト画面を前面に出すなら0
- 0（第5の0）：上下の向きがそのままなら0
- 0（第6の0）：左右の向きがそのままなら0

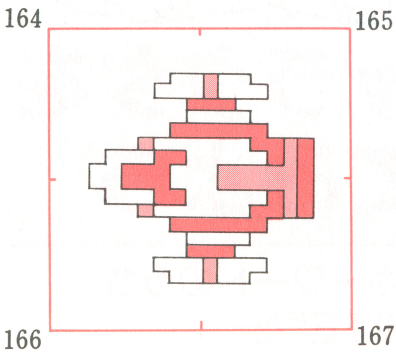


♠表示したいキャラクタは番号で指定して

30 DEF SPRITE 0,（0，1，0，0，0）とやっとここまではたどり着いた。と、そのあとがまたむずかしそうだな。このあとは、

CHR＄（164）＋CHR＄（165）＋CHR＄（166）＋ CHR＄（167）となっている。これは、表示させるアニメキャラクタのキャラクタ番号を指定している。

キャラクタテーブルAのスターシップのキャラクタを見てみよう。



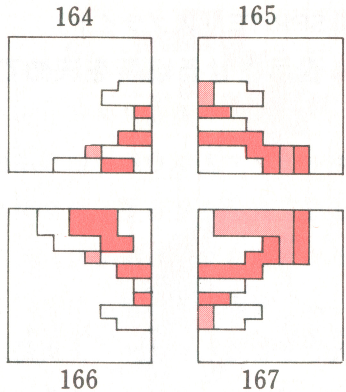
●ポイント

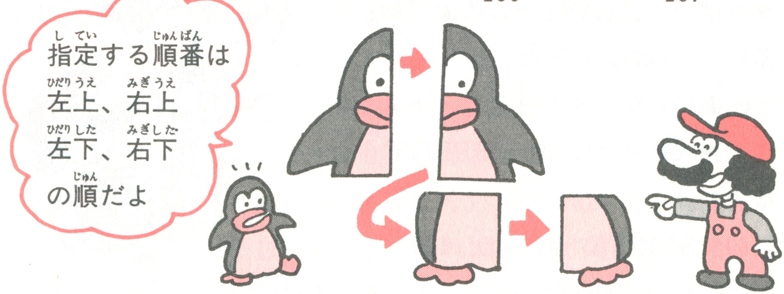
164、165、166、167がキャラクタ番号だ。

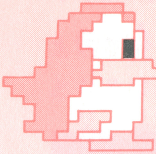
CHR＄は、キャラクタを指定するときの決まり文句だといっていい。

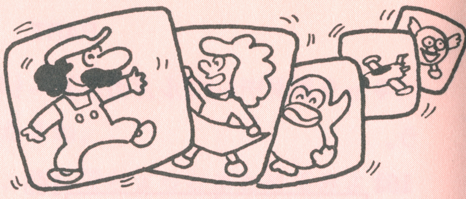
キャラクタを囲むわくの4すみに、164、165、166、167、という数字が書いてあるね。この数字がキャラクタ番号だ。

アニメキャラクタは、各8ドットずつの大きさの164、165、166、167という4個のキャラクタを合わせて作ってあるのだよ。









5.4 アニメキャラクタを表示させてみよう

SPRITEの3個のキーワードのうちしめくくりは"SPRITE"だ

■SPRITE（スプライト）

♠表示させる場所を決めてキャラクタを表示



スプライト画面にスイッチを入れて、キャラクタ表示OKという準備もすませた。表示させるキャラクタを選んで、その大きさや色も決めた。そこでいよいよ、そのアニメキャラクタをどこに表示させるかを決めて、実際に表示させればいい、というところまでたどり着いたわけだ。

画面にスプライト（アニメキャラクタ）を表示させるのが、SPRITE命令なのだ。

```
40　SPRITE　0，100，100
```

はじめのSPRITE　0は、スプライト番号0のキャラクタを表

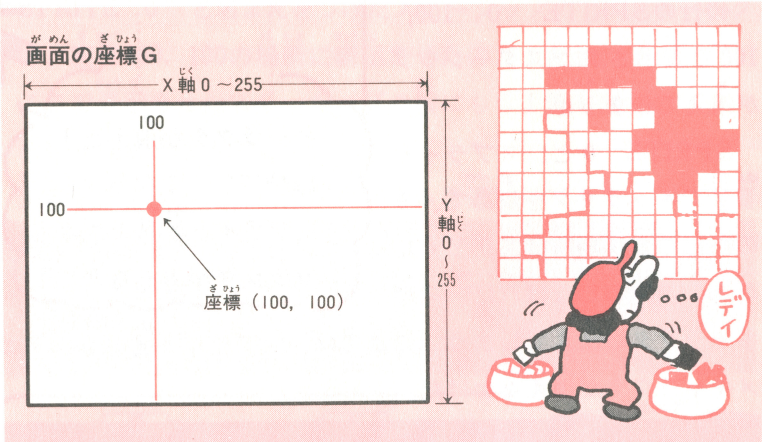
示しなさいということだ。

0につづいて、100, 100という数字が並んでいるが、これは、そのキャラクタを表示させる場所を示している。この100, 100は、テレビ画面の座標位置というんだ。

はじめの100はX軸の数で横の座標位置、次の100はY軸の数でたての座標位置だ。

●ポイント

SPRITE　0の0は、スプライト番号だが、これは、前にあったDEF SPRITE で指定したスプライト番号を使用するのだ。





キャラクタを表示させる場所（座標）は、X軸、Y軸とも0～255の数で指定することができるが、家庭用のテレビに表示させるには、X軸（横）が0～240、Y軸（たて）が5～220ぐらいで指定したほうがいい。それ以上の数で指定すると、キャラクタが画面からはずれることがあるからだよ。

♠SPRITEを使ってキャラクタを消してみる

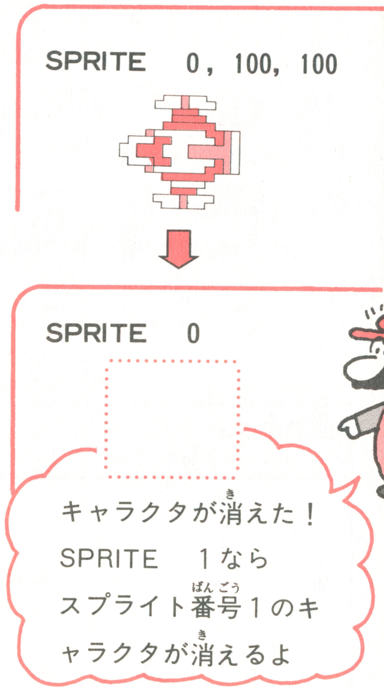
SPRITE命令は、決めた場所にアニメキャラクタを表示させる命令だといったが、実は、指定したアニメキャラクタを画面から消すこともできるのだ。

これまでのプログラムに、次の50行を追加してみよう。

```
50 SPRITE 0
```

40行のSPRITE　0，100，100で表示されたスターシップが消えてしまうね。

SPRITE　0と、スプライト番号でキャラクタを決めて、そのあとの表示場所（X，Y）になんの指定もしなければ、そのキャラクタは消えてしまう。ここでは、スプライト番号0のアニメキャラクタが消えたわけだ。

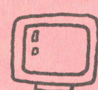




●ポイント

スプライトを横に並べて表示できるのは4個まで。4個以上表示させようとすると、そのうちの4個だけが表示されるよ。

2個のスプライトが同じ場所に重なったときは、スプライト番号の少ないほうが上になって重なって表示されるよ。





♠左向きと右向きのスターシップを表示させてみる

リスト1の「スターシップを表示させるプログラム」を、ここでもう一度復習してみよう。

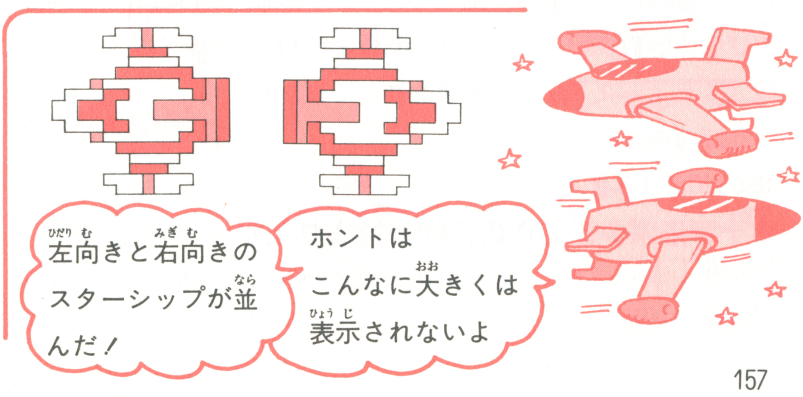
```
10　CLS
20　SPRITE　ON
30　DEF　SPRITE　0，（0，1，0，0，0）＝
　　CHR＄（164）＋CHR＄（165）＋CHR＄（166）＋
　　CHR＄（167）
40　SPRITE　0，100，100
```

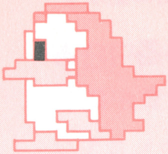
〝SPRITE　0〞と名まえをつけたスターシップのアニメキャラクタを、座標100，100の場所に表示させたのだったね。

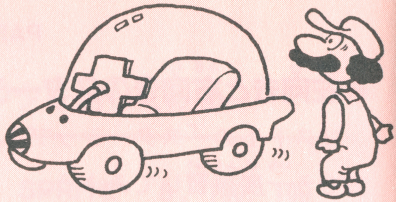
ここで表示させたスターシップの横にもう1個右向きのスターシップを表示させてみよう。

上のプログラムに、次のプログラムを追加してみる。

```
50　DEF　SPRITE　1，（0，1，0，1，0）＝
　　CHR＄（165）＋CHR＄（164）＋CHR＄（167）＋CHR＄
　　（166）
60　SPRITE　1，140，100
```





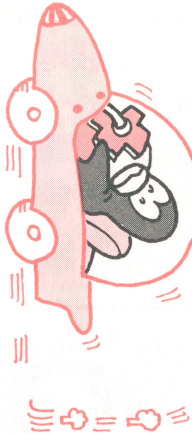


5.5 スターシップを動かしてみる前に

コントローラで早く動かしてみたいが その前に知っておかなければならないこと

♠コントローラがどの方向に押されたか？

画面に表示されたスターシップのアニメキャラクタを、コントローラで自由に動かしてみたいね。コントローラは、十字形のボタンやトリガボタンを押すたびに、信号をファミコンに送っているんだよ。



コントローラでアニメキャラクタを動かす前に、その信号を数値に変えてプログラムに呼びこむSTICK（スティック）関数とSTRIG（スティックトリガ）関数を知っておかなければならないよ。

■STICK（スティック）

♠十字形ボタンのどの方向が押されたか

まず十字形のボタンが、どの方向に押されたかを、STICK関数を使って調べてみよう。

次のプログラムを入力して、RUNさせる。

リスト2■十字形ボタンの押された方向を調べる

```
10  S=STICK（0）
20  PRINT  S
30  GOTO  10
```

●ポイント

このプログラムは、30行に行くと10行にもどり、また30行に行くというようにきりがない。

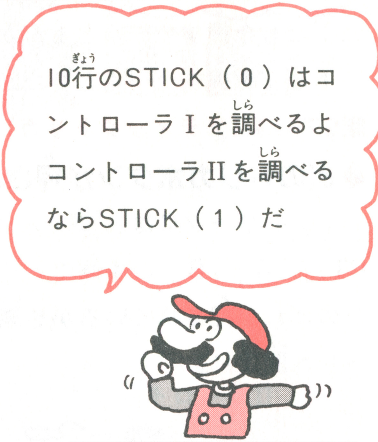
プログラムの実行を止めるのはSTOPキーだよ。

プログラムをRUNさせたら、コントローラの十字形ボタンの好きな方向を選んで押してみよう。

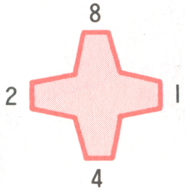
画面の左に数字がどんどん表示されていくね。

このSTICK関数は、十字形ボタンがどの方向が押されたかを調べて、数値に変える命令なんだよ。

画面に表字される数は、0、1、2、4、8の5種類。十字形ボタンのどの方向を押せば、どの数字が出るかは、次を見てほしい。



右を押す……1
左を押す……2
下を押す……4
上を押す……8
押さない……0





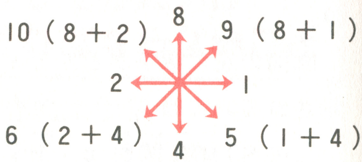
♠2つの方向を同時に押せば斜め方向だ

同時に2つの方向の十字形ボタンを押して、斜めの方向も得られるんだ。たとえば、右と上のボタンを同時に押して、右上の方向とか、上と左のボタンを同時に押せば、左上の方向だとか……。

こんなときにも、ファミコンは、ちゃんと判断して、その数値を示すよ。

右と上を押す……1と8だから9
左と上を押す……2と8だから10
下と右を押す……4と1だから5
下と左を押す……4と2だから6

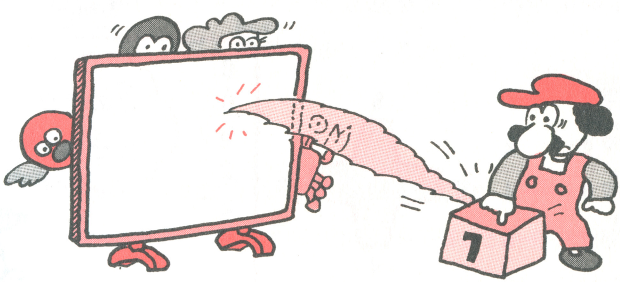
●ポイント
斜めの方向は、同時に押された2個のボタンの数を加えた数で示されるのだ。



■STRIG（スティックトリガ）

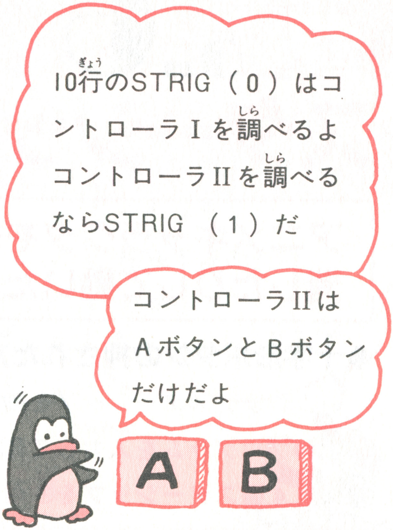
♠どのトリガボタンが押されたかを調べる

コントローラには、「スタート（START）」「セレクト（SELECT）」「A」「B」の4個のトリガボタンがある。このトリガボタンのどれが押されているかを調べて、数字で示すのがスティックトリガ関数だ。



リスト3■どのトリガボタンが押されたかを調べる

```
10  S＝STRIG（0）
20  PRINT  S
30  GOTO  10
```

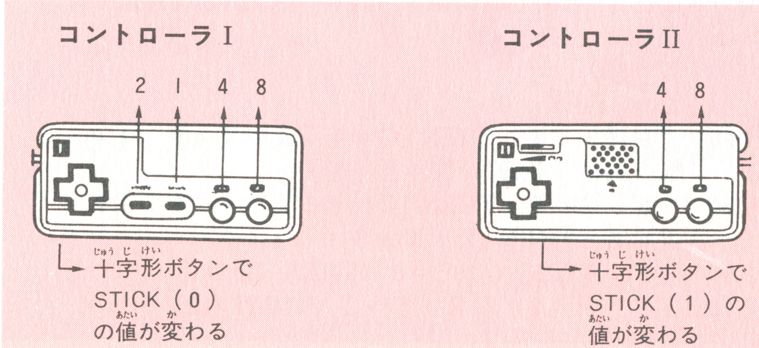


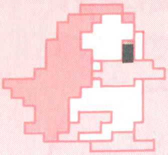
プログラムをRUNさせたら、コントローラⅠのトリガボタンのどれでもいいから押してみよう。

押したボタンごとにそれぞれ、1、2、4、8のいずれかの数字が、画面に表示されたはずだ。なにも押していなければ0だ。

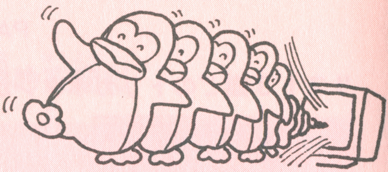
STARTボタン……1
SELECTボタン…2
Aボタン………………8
Bボタン………………4
なにも押されない……0

と、ここまではあまりおもしろくなかったかもしれないが、これを頭において、次はいよいよ、コントローラでスターシップを動かすプログラムにGO! だ。





5.6 スターシップが動いた



アニメキャラクタを表示させたり消したりして動いたようにみせる

♠十字形ボタンの押された方向に動くように

いよいよ、スターシップを動かしてみよう。

次のプログラムを入力してRUNさせてみよう。

リスト4■スターシップを左右に動かす

```
10  CLS
20  SPRITE ON
30  DEF SPRITE 0,(0,1,0,0,0)=
    CHR$(164)+CHR$(165)+CHR$(166)+
    CHR$(167)
40  DEF SPRITE 1,(0,1,0,1,0)=
    CHR$(165)+CHR$(164)+CHR$(167)+
    CHR$(166)
50  X=120
60  S=STICK(0)
70  IF S=1 THEN C=1:X=X+8:IF
    X>240 THEN X=X-8
80  IF S=2 THEN C=0:X=X-8:IF
    X<5 THEN X=X+8
90  IF STRIG(0)=8 THEN END
100 IF S=0 OR S>2 THEN 120
```

```
110 SPRITE 0：SPRITE 1
120 SPRITE C, X, 100
130 GOTO 60
```

これで、コントローラを押せば、表示されたスターシップが左右に動くようになっている。

プログラムに出てくる命令は、ここまでにべんきょうずみのものだけだから、わかるはずだけど、かんたんにプログラムのポイントを説明しておこう。

ここに出てくる命令は
みんなもう
勉強ずみのもの
だから
わかるよね

■プログラムのかんたんな説明

左向きと右向きのスターシップを表示させて、これを、コントローラの押された方向で、左右に動かすのだが、さて、プログラムの要点は——。

10～50行　スプライトが表示できるようにし、右向きと左向きのスターシップを定義し、最初に表示するX軸の座標を120とする。

60行　十字形ボタンのどの方向が押されたかを調べる。

70行　十字形ボタンの右が押されたらスターシップを8ドット右に表示する。Cはスプライト番号。Xが240より大きいと画面からはみ出



右方向のボタンを
押すというのは
S＝1ということだ

るので、X－8としている。

80行　スターシップを8ドット左に表示する。

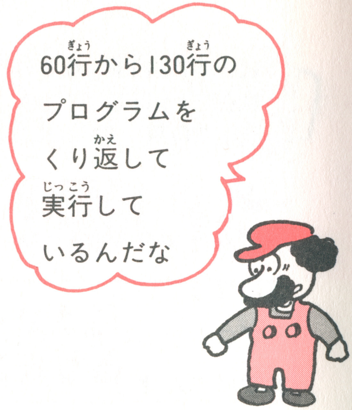
90行　トリガボタンAを押せばEND。

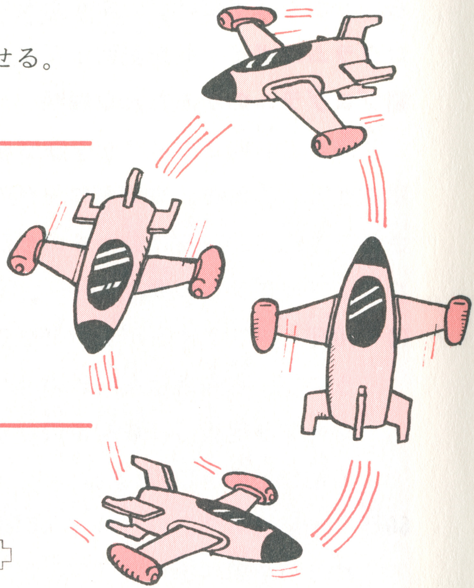
100行　十字形ボタンの上下方向が押されるか、ボタンが押されなかったら、110行を飛び越えて120行へ。

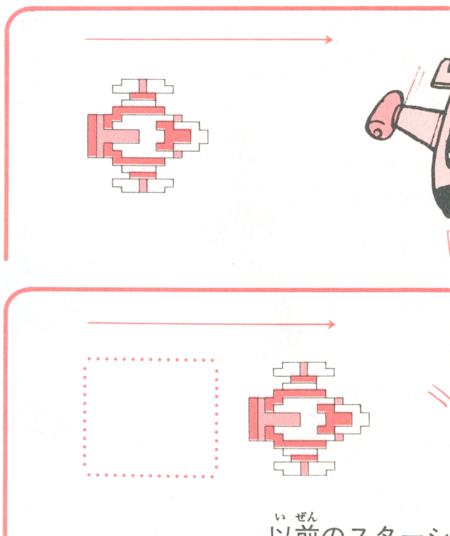
110行　スプライト番号0と1のスターシップを消す。

120行　スターシップを表示させる。

130行　60行にもどる。







以前のスターシップが消えて、8ドット右に表示されれば、見た目には右に動いたように見えるよ。

♠スターシップの上下左右移動に挑戦だ

スターシップを左右に動かすプログラムを応用すれば、上下左右に動かすプログラムも、かんたんにできるとは思わないかい？

左右に動かすプログラムは、X軸の数値を変えたのだから、上下左右に動かすなら、Y軸の数値を変えてやるプログラムを追加してやればいい、ということになるんだろう。



ここでまた少し、ヒントをあげるから、リスト4「スターシップを左右に動かす」プログラムを改造して、上下左右に動くプログラム作りに挑戦だ。



☆応用プログラム作りのヒント

ヒント1　40行と50行のあいだに、上向きと下向きのスターシップのスプライトパターンを定義する。

ヒント2　80行と90行のあいだに、十字形ボタンの押された方向しだいで、Y軸の値が変化するプログラムを入れる。

ヒント3　100行は、十字形ボタンが押されたときだけ、120行に行くようにする。

ヒント4　新しく定義したスターシップも消す。

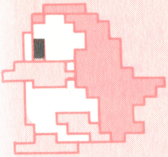
ヒント5　スターシップを表示するY軸の値を変数で指定する。

さあ、どうだ、できたかな？　リスト5に、そのプログラムを

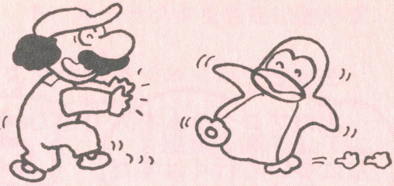
あげておくから、できた人もできなかった人も参考にしてほしい。

リスト5■スターシップを上下左右に動かす

```
10  CLS
20  SPRITE ON
30  DEF SPRITE 0, (0, 1, 0, 0, 0) =
    CHR$ (164) +CHR$ (165) +CHR$ (166) +
    CHR$ (167)
40  DEF SPRITE 1, (0, 1, 0, 1, 0) =
    CHR$ (165) +CHR$ (164) +CHR$ (167) +
    CHR$ (166)
43  DEF SPRITE 2, (0, 1, 0, 0, 0) =
    CHR$ (172) +CHR$ (173) +CHR$ (174) +
    CHR$ (175)
46  DEF SPRITE 3, (0, 1, 0, 0, 1) =
    CHR$ (174) +CHR$ (175) +CHR$ (172) +
    CHR$ (173 )
50  X =120 : Y =100
60  S =STICK (0)
70  IF S = 1  THEN C = 1 : X = X + 8 : IF
    X >240  THEN X = X - 8
80  IF  S = 2  THEN C = 0 : X = X - 8 : IF
    X < 5  THEN X = X + 8
83  IF  S = 4  THEN C = 3 : Y = Y + 8 : IF
    Y >220  THEN Y = Y - 8
86  IF  S = 8  THEN C = 2 : Y = Y - 8 : IF
    Y < 5  THEN Y = Y + 8
90  IF STRIG (0) = 8  THEN END
100 IF  S = 0  THEN 120
110 SPRITE 0 : SPRITE 1 : SPRITE
    2 : SPRITE 3
120 SPRITE C, X, Y
130 GOTO 60
```



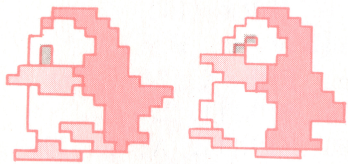
5.7



ペンペンくん歩いてね

**アニメをただ動かすだけでなく
歩かせるというところに変化がある**

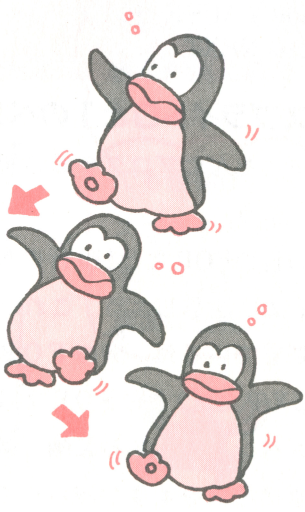
♠2個のキャラクタを消して描いて消して描いて



ペンペンくんをモデルに使って、アニメキャラクタを歩かせてみようというわけだ。ただ、左や右に動くのとちがって、歩くのは足を交互に出すのだから、それだけ複雑になるのは当然だよ。

アニメキャラクタを、どうやって歩いているようにみせるかという問題だ。

右の2枚のアニメキャラクタ・ペンペンくんを見てほしい。この2枚のペンペンくんを交互に表示させて、歩いているようにみせるというわけだよ。



さっそくペンペンを歩かせようじゃないか。

次のプログラムを入力して、RUNさせよう。ペンペンが右から左に歩いて行くはずだ。

リスト6 ■ペンペンを右から左に歩かす

```
10  CLS
20  SPRITE ON
30  DEF SPRITE 0,(0,1,0,0,0)=
    CHR$(96)+CHR$(97)+CHR$(98)+CHR$
    (99)
40  DEF SPRITE 1,(0,1,0,0,0)=
    CHR$(100)+CHR$(101)+CHR$(102)+
    CHR$(103)
50  A=0:B=1
60  FOR X=240  TO 0  STEP -4
70  SPRITE B
80  SPRITE A,X,100
90  SWAP A,B
100 PAUSE 10
110 NEXT
```

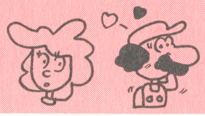
♠スプライト0と1のペンペンを交互に表示

かんたんにプログラムの説明をしよう。

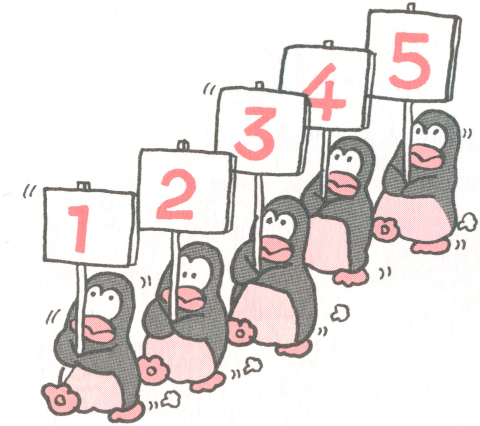
60行のFOR文と110行のNEXT文に注目しよう。これはもう知っているね。例のFOR～NEXT文だ。このFOR文は、ペンペンを表示するX軸（横）の座標位置を4ずつ減らしてい

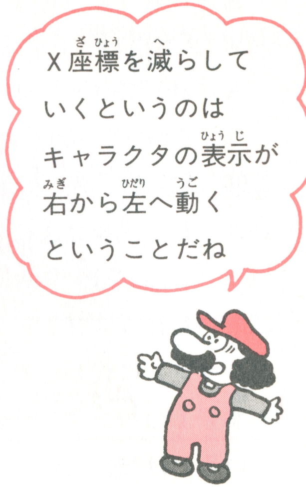
●ポイント

FOR～NEXTはFORとNEXTにはさまれた部分をそこで指定する回数だけくり返すのだったね。



く命令だ。

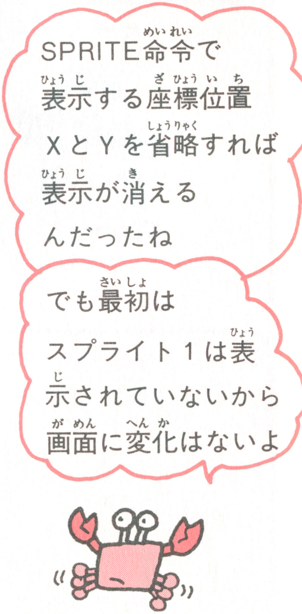


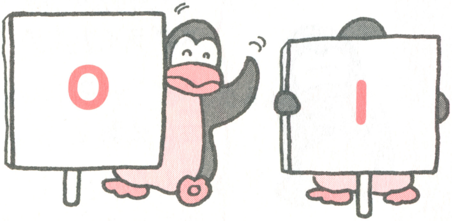


ペンペンを歩かせるためのプログラムは、70行～90行だ。まず70行からみていこう。

70　SPRITE　B

変数Bで指定したペンペンを消している。変数Bとはなにか、というと、50行にA＝0：B＝1とあるから、スプライト番号1のペンペンを消すというわけ。





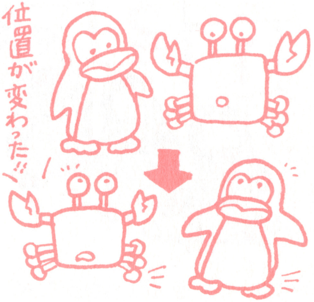
変数Aのペンペンを表示するのが80行だ。

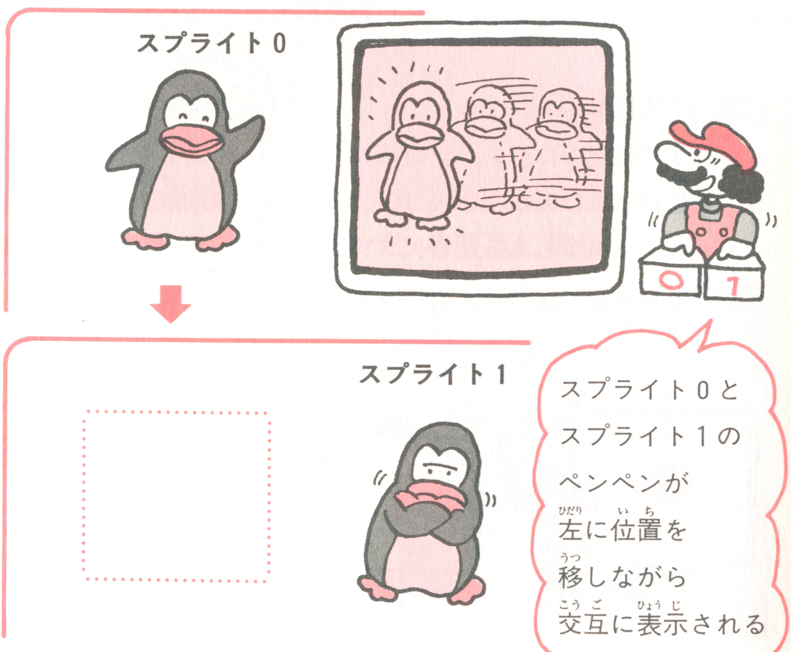
```
80 SPRITE A, X, 100
```

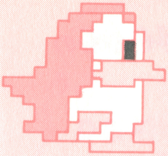
Aにはスプライト0が代入されているから、スプライト0のペンペンを（X，100）の座標に表示させる。

```
90 SWAP A, B
```

AとBを交換せよという命令だから、こんどは、これまで表示されていたB（スプライト番号0のペンペン）に代わって、スプライト番号1のペンペンが表示されるというわけだ。







5.8 もっとかんたんにペンペンが歩いた



**かんたんなことはいいことだ
そんな動かしかたを教えてしまおう**

♠DEF MOVEとMOVEを使って動かす

スプライトを定義して、アニメキャラクタを動かす方法を説明してきたが、実は、もっとかんたんに、キャラクタを動かすこともできるのだ。ペンペンを歩かせてみようか。

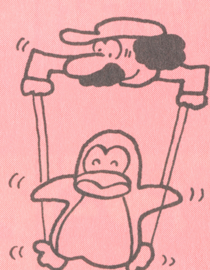
ペンペンを歩かせるのはMOVE 命令だ。

MOVE命令は、表示させるアニメキャラクタと、その動く方向を決めて、かんたんにアニメキャラクタを動かしてしまうよ。

次のプログラムを入力して実行だ。このプログラムは、リス

●ポイント

MOVE命令は2個のキャラクタを交互に表示させる。マリオやペンペンを動かすときも、自動的に2個のキャラクタを交互に表示させてくれるよ。

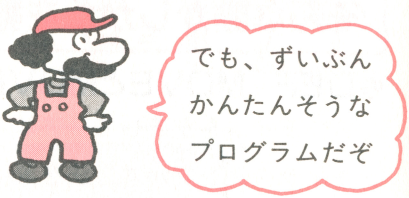


ト6の「ペンペンを右から左に歩かす」と同じ動きをさせるものだ。

リスト7 ■MOVE命令でペンペンを歩かす

```
10  CLS
20  SPRITE  ON
30  DEF  MOVE（0）=SPRITE（4, 7, 4,
    120, 0, 0）
40  POSITION  0, 240, 100
50  MOVE  0
```

このプログラムの要点は、30行のＤＥＦ　ＭＯＶＥ命令と50行のＭＯＶＥ命令だ。これらをゆっくり説明していこう。



■DEF MOVE（デフ　ムーブ）

♠どのキャラクタをどの方向にどれだけ動かすかだ

表示して動かすキャラクタを決め、どの方向に動かすかなどから始めなければならない。これを決めて指示するのが、DEF　MOVE命令だ。

30行を見てみよう。

```
30  DEF  MOVE（0）=SPRITE（4, 7, 4,
    120, 0, 0）
```



このDEF　MOVE命令を、次のように書きかえて説明しょう。

DEF　MOVE（n）=SPRITE（A，B，C，D，E，F）

n…その動作を番号で指定する動作番号。0～7の数字を使う。

A…表示させるアニメキャラクタの番号。0～15の数字を使う。

キャラクタ番号は、次のようになっているよ。

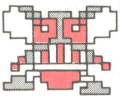
0　マリオ



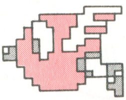
1　レディ



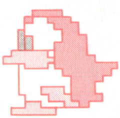
2　ファイターフライ



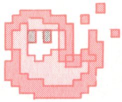
3　アキレス



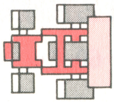
4　ペンペン



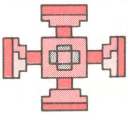
5　ファイアーボール



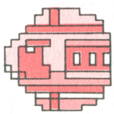
6　車



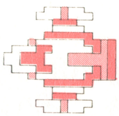
7　スピナー



8　スターキラー



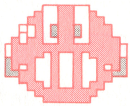
9　スターシップ



10　爆発



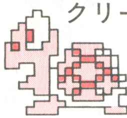
11　ニタニタ



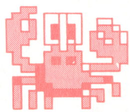
12　レーザー



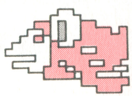
13　シェルクリーパー

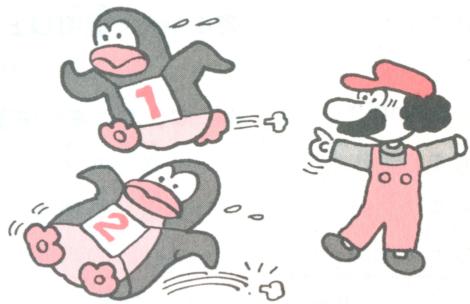


14　サイドステッパー



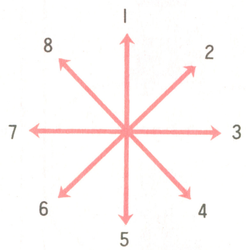
15　ニットピッカー





動作番号とは
たとえば（1）は
「1と名づけた
動作だよ」（2）は
「2と名づけた
動作だよ」という
意味だ

B…キャラクタを動かす方向。0～8の数字で指定。動かさなければ0。



C…アニメキャラクタを動かす速度。1～255の数で指定する。60ドット動くのにかかる秒数で指定すると考えていい。たとえば6なら、60ドット動くのに6秒だ。

数字が大きければ
動く速さが
遅いというわけ

120ドット動かしたい
ときは
60を指定する

D…アニメキャラクタを動かそうとする全移動距離。1～255の数で指定する。指定した数字×2ドットが移動距離だ。

E…アニメキャラクタをバックグラウンド面の背景より前面に表示させたいなら0、後面に表示させるなら1。

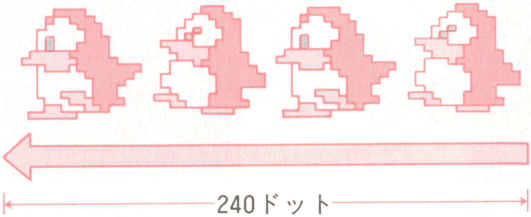
F…アニメキャラクタの配色番号。0～3の数で指定する。

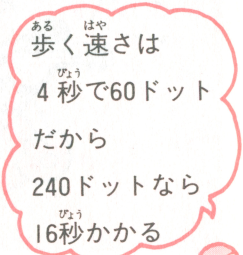


以上の指定を見て、30行のDEF MOVE命令を読みとってみよう。

まず、DEF　MOVE（0）だから、この動作は0と名づけているね。

SPRITE（4，7，4，120，0，0）を分解すれば、表示させるキャラクタはペンペン（4）、右から左の方向（7）に動かすのだが、その速度は4秒で60ドットだ。全移動距離は240ドット、アニメの表示を背景より前に出し、配色番号0の配色にする、とこうなる。







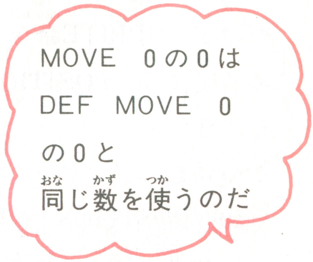
■MOVE（ムーブ）

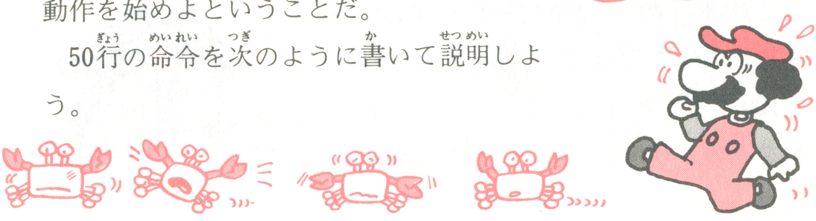
♠別な動作を8つまで同時にさせることができる

DEF　MOVE命令で、どんなキャラクタを、どのように動かすかを決めたね。その動き方を、実際に動かすのがMOVE命令だ。

50行はMOVE　0とあるね。これは、動作を始めようということだ。

50行の命令を次のように書いて説明しよう。

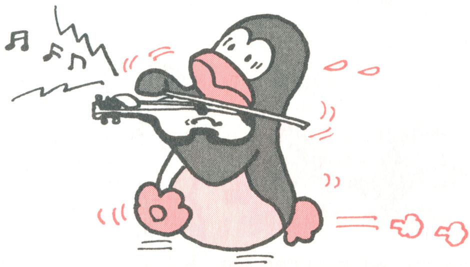




MOVE　n_0 [,n_1,n_2,n_3,n_4,n_5,n_6,n_7]

n_0からn_7までは、DEF　MOVEで定義するキャラクタの動作だ。同時に8種類の動作までさせることができるのだよ。

たとえば、MOVE　2，4，6なら、動作番号2，4，6と名づけられた動作が、同時に始まるよ。



●ポイント

MOVE命令でキャラクタを動かしているとき、音楽演奏するなら、テンポは2倍だよ。

■POSITION（ポジション）

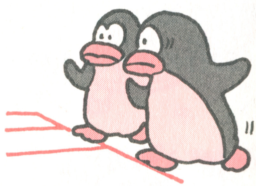
♠ここの位置から動き始めなさいという命令だ

アニメキャラクタを動かすのに、MOVE命令はたいへんべんりだけど、SPRITE命令を使ったときとちがって、表示して動き始める座標は、POSITION命令で指定しなければならないから注意しよう。

40行のプログラムを見てみよう。

```
40　POSITION　0，240，100
```

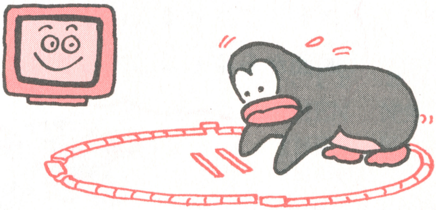
これは、動作番号0の動作を開始する座標位置は、X軸（横）が240、Y軸（たて）が100のところから、という命令だ。これを次のように書いて、くわしく説明しよう。



POSITION　n, X, Y

n,…動作番号だから、DEF　MOVEで指定したのと同じ動作番号となる。

X, Y…n,の動作をさせるキャラクタが、動作を開始する座標位置だが、指定範囲は0～255までだ。



●**ポイント**

動作を始める位置は、たて、横とも0～255まで指定できるが、あまり端を指定すると画面からとび出してしまうから実際にはXを0から240、Yを5～220の範囲で指定してくださいね。

■CUT（カット）

♠動きだしたキャラクタを一時ストップ！

アニメキャラクタを動かすMOVE命令に関係した命令では、このほかにCUT命令やERA（イレーズ）命令があるよ。

CUT命令は、MOVEでスタートしたキャラクタの動作を一時的にストップさせるんだ。ストップさせる動作は、n_0からn_7まで8種類の動作だよ。



CUT　n_0 [, n_1, n_2, n_3, n_4, n_5, n_6, n_7]

n…0～7までの数で指定した、その動作の動作番号。たとえば、

動作番号0と3で指定した動作をストップさせるなら、

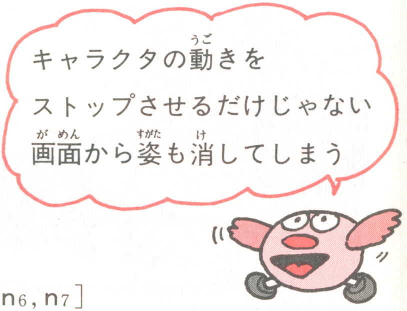
CUT　0，3

とするんだよ。

■ERA（イレーズ）

♠動作を止めて表示も画面から消してしまう

CUT命令は、動いているアニメキャラクタを動かなくするだけだが、ERA命令は、キャラクタの動きをストップさせたうえで、画面から消してしまうのだよ。

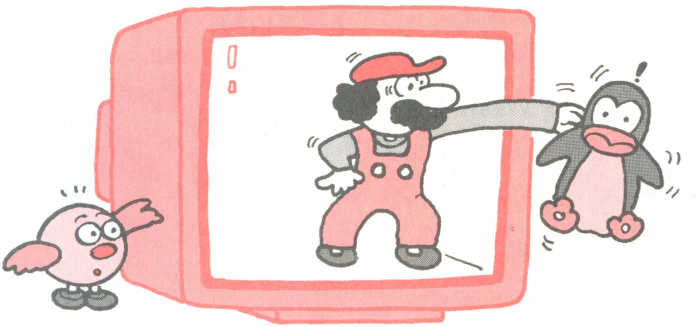


ERA　n_0 [,n_1,n_2,n_3,n_4,n_5,n_6,n_7]

n…0～7までで指定した動作のうち、ストップさせて消したい動作の動作番号。たとえば、動作番号1、3の動作をストップさせて、画面から消したかったら、

ERA　1，3

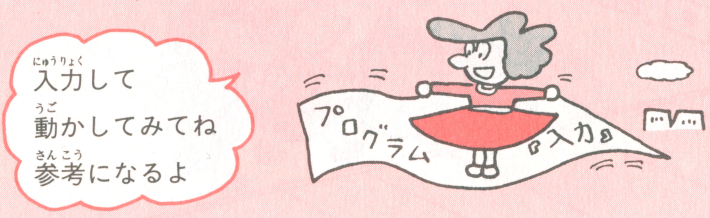
とするんだよ。

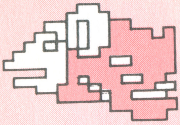


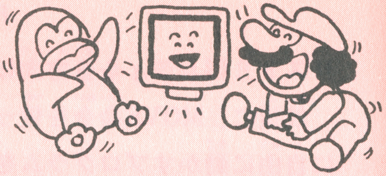
リスト8 ■レディを全方向に動かす

最後に、MOVE命令を使って、レディを左右上下斜めの8方向に自由に動かすプログラムをのせておこう。

```
10  CLS
20  SPRITE ON
30  FOR I=0 TO 7
40  DEF MOVE(I)=SPRITE(1,I+1,1,3,
    0,0)
50  NEXT
60  MOVE 0
70  S=STICK(0):IF S=0 GOTO 60
80  IF S=1 THEN N=2:GOTO 160
90  IF S=2 THEN N=6:GOTO 160
100 IF S=4 THEN N=4:GOTO 160
110 IF S=8 THEN N=0:GOTO 160
120 IF S=9 THEN N=1:GOTO 160
130 IF S=5 THEN N=3:GOTO 160
140 IF S=6 THEN N=5:GOTO 160
150 IF S=10 THEN N=7
160 IF MOVE(M)=-1 GOTO 160
170 IF M=N THEN MOVE M:GOTO 70
180 ERA M
190 POSITION N,XPOS(M),YPOS(M)
200 M=N
210 MOVE N:GOTO 70
```







5.9 オリジナルゲームにアタック

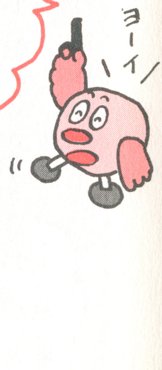
**背景の作り方もアニメの動かし方も
もうみんなわかったら、ゲーム作りにGO！**

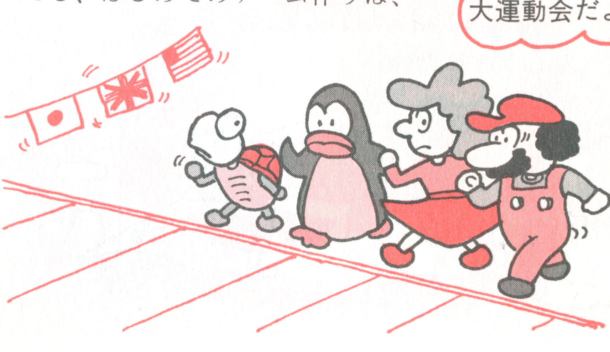
♠アニメキャラクタの大運動会ゲームだ

背景の作り方もおぼえたし、ベーシックの勉強もすんだ。ファミコンのアニメキャラクタも、上下左右に自由に動かせるようになったとしたら、あとは自分だけのオリジナルゲーム作りにアタックするだけだ。

でも、はじめてのゲーム作りは、

ゲームは
レディ、マリオ、ペンペン、カニさん、カメさんの
2人と3びきの
大運動会だよ





頭ではわかっていても、いざ実際にプログラミングとなると、どこから手をつけていいいか、わからないものだよ。だから、ここでひとつの参考プログラムを作ってみせて、かんたんに説明しておくから、そのプログラムを入力して遊んだり、プログラミングの勉強もしてほしい。

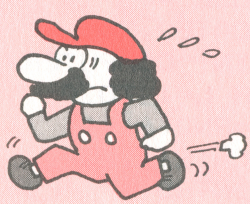
このゲームの題名は「ファミコンキャラクタ運動会」とでもしておこう。

アニメキャラクタのマリオ、レディ、ペンペン、シェルクリーパー、サイドステッパーの2人と3びきが、サイコロの数だけ進む大運動会だ。

まず、はじめに、プログラムを示しておくよ。

リスト9■ファミコンキャラクタ運動会

```
10 CLS:CGSET1,1:CGEN2
20 SPRITE ON
30 DEF MOVE(1)=SPRITE(0,7,1,2,0)
40 DEF MOVE(2)=SPRITE(1,7,1,2,0)
50 DEF MOVE(3)=SPRITE(4,7,1,2,0)
60 DEF MOVE(4)=SPRITE(13,7,1,2,0)
70 DEF MOVE(5)=SPRITE(14,7,1,2,0)
80 DIM NA$(5)
90 LOCATE 0,2
100 FOR I=1 TO5
110 PRINT I;"ハ゛ンメノヒトノナマエヲイレテクタ゛サイ"
120 PRINT
130 INPUT" ";NA$(I)
140 PRINT
150 NEXT
160 CLS
170 VIEW
180 FOR I=1 TO5
```



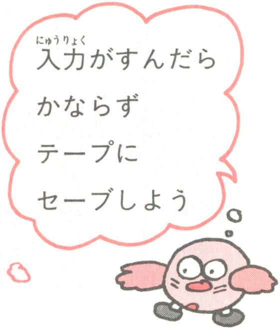
```
190 FOR J=0 TO4
200 POSITION I,220-4*J,24+24*I
210 MOVE I
220 IF MOVE(I)=-1 GOTO220
230 NEXT
240 NEXT
250 FOR K=1 TO5
260 IF NA$(K)="" GOSUB 470:GOTO 280
270 GOSUB 380
280 FOR L=1 TORD
290 X=XPOS(K)
300 POSITION K,X,24+24*K
310 MOVE K
320 IF MOVE(K)=-1 GOTO 320
330 NEXT
340 BEEP
350 IF XPOS(K)<=40 THEN W=K:K=5:NEXT:GOTO 560
360 NEXT
370 GOTO 250
380 LOCATE2,20
390 PRINTLEFT$(NA$(K),10);"ｻﾝﾉ ﾊﾞﾝﾃﾞｽ            "
400 LOCATE13,22:PRINT"  "
410 IF STRIG(0)<>8 GOTO 410
420 RD=RND(6)+1
430 LOCATE13,22
440 PRINTRD
450 IF STRIG(0)<>4 GOTO 420
460 RETURN
470 PAUSE 40
480 LOCATE2,20
490 PRINT"COMPUTERﾉ ﾊﾞﾝﾃﾞｽ          "
500 FOR L=1 TO20
510 RD=RND(6)+1
520 LOCATE13,22
530 PRINTRD
```

```
540 NEXT
550 RETURN
560 A1$="O2C1O3CECGECEO2CAO3DO2A"
570 A2$="O3FDO2AO3DO2GBO3BAG#FFD"
580 A3$="O2BO3CECO2BGFD"
590 A4$="O2CO1BO2ECGEO3C5O2R"
600 PLAYA1$:PLAYA2$:PLAYA3$
610 PLAYA1$:PLAYA2$:PLAYA3$
620 PLAYA4$
630 CLS
640 ERA 1,2,3,4,5
650 IF NA$(W)="" THEN W$="COMPUTER":GOTO670
660 W$=NA$(W)+"サン"
670 LOCATE 5,10
680 PRINTW$;"ノ カチデス"
690 LOCATE 5,13
700 PRINT"モウイチド ヤリマスカ ?(Y/N)
710 LOCATE 5,15
720 INPUT K$
730 IF K$="Y" THEN RUN
740 IF K$="N" THEN END
750 GOTO 720
```

■ゲームの背景画作り

♠背景画はゲームのコースだ。なくても遊べるよ

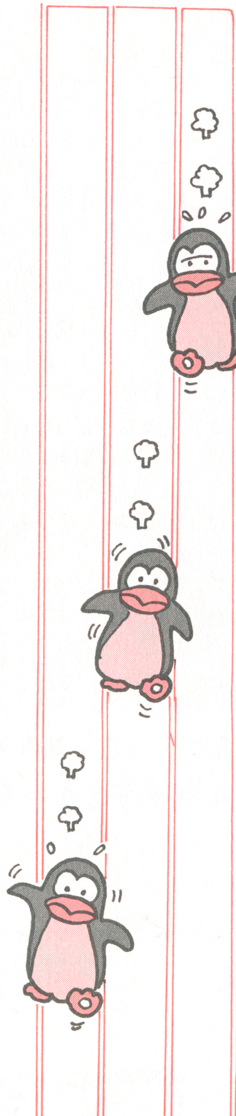
まず、BGグラフィックを使って背景画作りからだ。背景画のデータは、次のページにのせておくけれど、入力するのがめんどうだったら、背景画がなくてもゲームは、できるよ。





（コース作りデータ表）

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1																												
2	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62
3				J22																				J22				
4				J22																				J22				
5	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62
6				J22																				J22				
7				J22																				J22				
8	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62
9				J22																				J22				
10				J22																				J22				
11	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62
12				J22																				J22				
13				J22																				J22				
14	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62
15				J22																				J22				
16				J22																				J22				
17	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62	L62
18																												
19																												
20																												

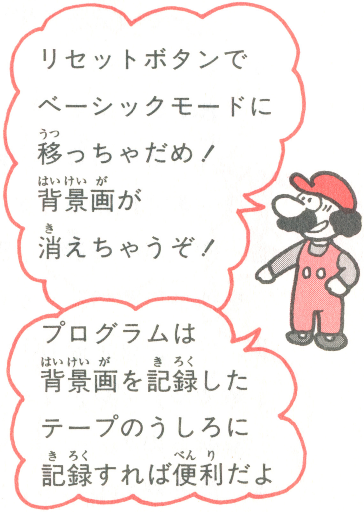


■プログラムの入力

♠うっかり背景画を消してしまわないように

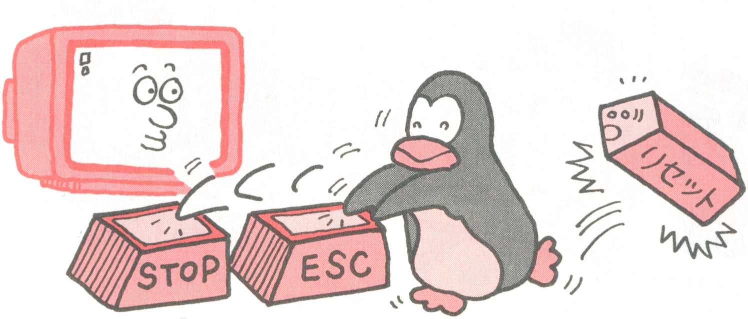
背景画の入力がすんだら、次はいよいよプログラムの入力だ。ベーシックモードに移すのだが、ここでちょっと待った。だいじなことがあるんだ。かならず守ってほしいことだ。

リセットボタンを押して、ベーシックモードに移ると、BGグラフィックのせっかく作った背景画が消えてしまうぞ。だからかならず、ESCキーを押してから、STOPキーを押すことだ。



そのあとで、ゆっくり、リスト9をプログラム入力すればいい。あわてるな。プログラムは、どこにも逃げていかないよ。

プログラムの入力がすんだら、RUNさせて、入力ミスがないかどうか、たしかめよう。入力ミスがなかったら、テープにセーブだ。

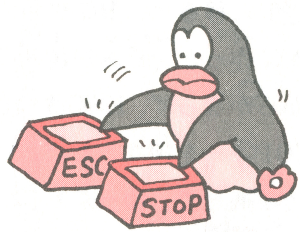


♠プログラムの動かし方をもういちど確認しよう

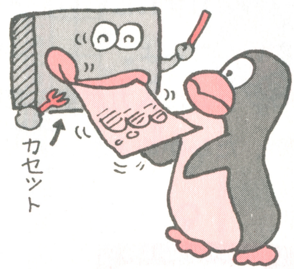


もういちど、プログラムの動かし方と、その順序を確認しておこう。

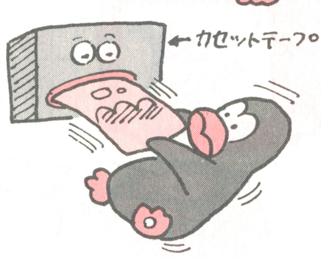
1．BGグラフィックで、テープに入っている背景画のデータをロード（LOAD）する。



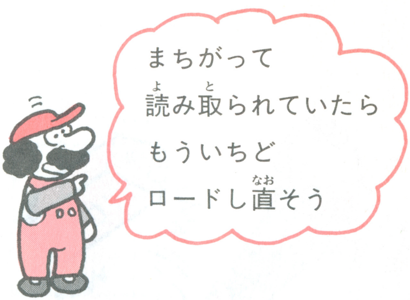
2．ESCキーを押し、STOPキーを押して、ベーシックに移る。



3．ベーシックのLOAD（ロード）命令で、プログラムをテープから読み出す。



4．読み終わったら、LIST（リスト）命令で、プログラムを表示させるか、RUNさせてみて、正しく読み取られているかどうかをたしかめる。



■ゲームの遊び方

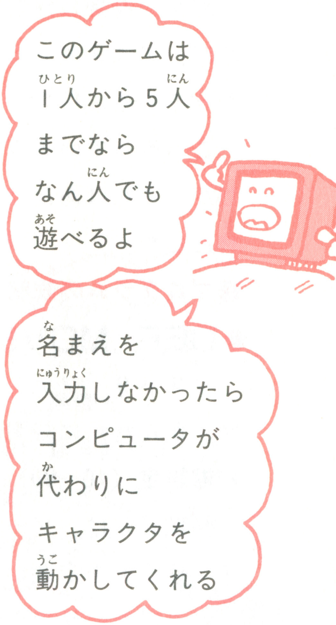
♠コンピュータが出すサイコロの数だけ進む

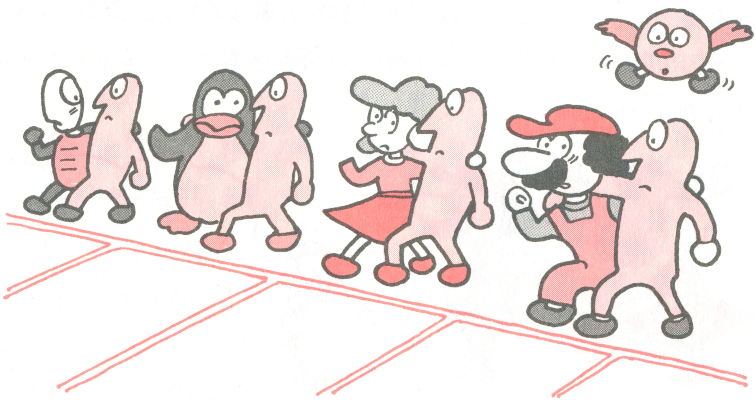
プログラムをRUNさせると、だれが、どのアニメキャラクタを動かすか聞いてくる。キャラクタは、マリオ、レディ、ペンペン、シェルクリーパー、サイドステッパーの順で、それを動かす人の名まえを聞いてくるのだよ。そうしたら、キーボードから、その人の名まえを入力してやろう。

入力する名まえは、ローマ字でもカタカナでもかまわない。

競技者がその名まえを全員入力し終わると、2人と3びきのキャラクタが、スタートラインに並んでスタンバイだ。

下に動かす人の名まえが表示されるから、その人はコントローラIのAボタンを押す。





すると画面には、1から6までの数（乱数）が次つぎに表示されるから、こんどはBボタンを押すんだ。

ボタンを押したときに表示されていた数だけが残るけど、その数が、アニメキャラクタの進む数だ。その数だけ左に進むことになっている。

こうやっているうちに、いちばん早くゴールインした人が勝ちだが、これはあたりまえだね。1人ゴールインすれば、ゲームは終わりだ。

■プログラムの説明をかんたんに

♠DEF MOVEでキャラクタの動きを決める

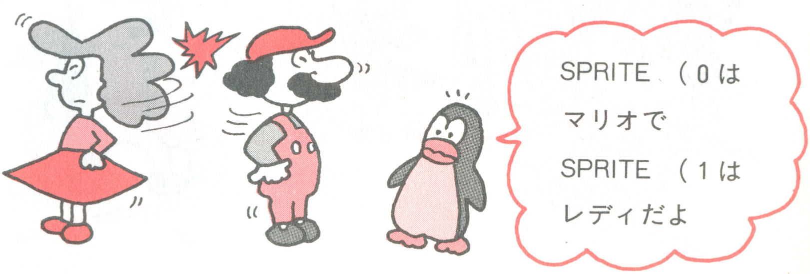
リスト9「ファミコンキャラクタ運動会」のプログラムの要点を、かんたんに説明しておこう。

初期設定（10〜80）

初期設定は、ゲームの中で使う変数に一定の値を代入したり、スプライトを定義しておいたりすることだ。

ここでたいせつなのは、30〜70行。DEF MOVE文でキャラクタの動き方に名まえをつけ、その動き方を決めている。

SPRITEの次の最初の数字は、表示させるアニメキャラクタの番号だよ。



5つの名まえを入力する（90〜150）

遊ぶ人の名まえを入力させる部分だ。5人の人の名まえを入れさせるのだから、同じことを5回くり返せばいい。そう、FOR〜NEXT文を使えばいいね。

名まえをキーボードから入力するのは、130行の

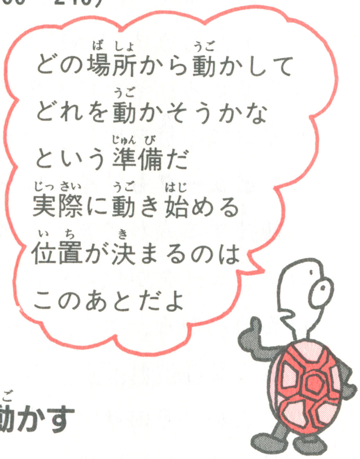
INPUT〝 〞；NA＄（I）

だよ。

どこの場所からでも動かせるように（160〜240）

画面をきれいにして（160 CLS）、BGグラフィックの内容を画面に表示したあと、200行のPOSITION命令は、キャラクタがどの場所からでも動かせるようにしている。いわば競争のスタートラインだ。

そして、210行のMOVE命令で、もう動けるよというわけ。



♠動作位置を決めてキャラクタを動かす

いよいよメインルーチン（250〜370）

いよいよメインルーチンに入る。まず、290行と300行。動作番号Kのキャラクタの X 座標位置を調べ、そこを動作開始の位置とするんだ。そして、310行のMOVE Kで、動作番号Kのアニメキャラクタを実際に移動させるというわけだ。



サイコロの数を出すサブルーチン(380〜460)

ここは、いつもメインルーチンに呼び出されるところで、サイコロの数を決めるサブルーチンだ。サイコロの目を出すのは、

420　RD＝RND（6）＋1

というところ。

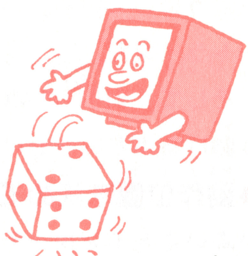
RND関数は知っているね。でも、ほんとうにわかるかな。なぜ、RND（6）に＋1するのだろう？　RND（6）は0〜5の数を出すから、それに1を加えて、1〜6のサイコロの目を出しているのだよ。



コンピュータもサイコロを振る(470〜550)

競技する人が名まえを書かなかったら、コンピュータが代わりにゲームを進めてくれるのだから、コンピュータにもサイコロを振らせなければならないね。

ここが、そのサブルーチンだ。やはり、RND（6）＋1だが、ここでは、20回目に発生した乱数をサイコロの目としている。



ゴールインしたら音楽を鳴らす(560〜750)

560〜620行はゴールインしたときのお祝いのファンファーレ。べつにむずかしいところはないはずだね。

720〜750は、再ゲームをするかどうかの答えを求めているんだよ。おしまいだ。



★ファミリーコンピュータ、ファミリーベーシックは
任天堂の商標です

この本の内容についての問い合わせは、
株式会社　新星出版社　☎03-831-0743
ファミリーコンピュータ係までお願い
いたします。

ファミリーコンピュータ™

ぼくらのファミコン入門

1986年1月15日　初版発行　©

著　者　山　下　利　秋
発行者　富　永　弘　一
印刷所　慶昌堂印刷株式会社

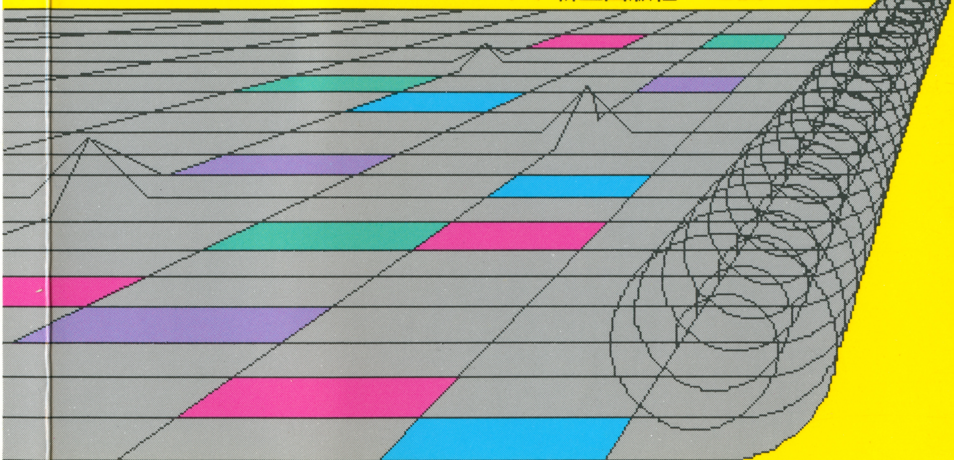
発行所　東京都台東区 台東2丁目24　株式会社　新星出版社
郵便番号110　電話(831)0743　振替東京 4-72233

ISBN4-405-06067-3

ファミリーベーシックでオリジナルゲームに挑戦

カセットゲームも楽しいけれど、ファミリーコンピュータにキーボードを接続すれば、キミだけのコンピュータランド。BGグラフィックできれいな背景をつくり、ゲームベーシック命令をマスターして、マリオやペンペンが動きまわるオリジナルゲームに挑戦しよう

ぼくらのファミコン入門 ★新星出版社 定価780円



ファミリーコンピュータ、ファミリーベーシックは任天堂の商標です。

ISBN4-405-06067-3 C2055 ¥780E

ファミリーコンピュータ™ ファミリーベーシック™

ぼくらのファミコン入門

ファミリーベーシックでオリジナルゲームに挑戦

★新星出版社

ファミリーコンピュータ™
ファミリーベーシック™

ぼくらのファミコン入門

ファミリーベーシックで
オリジナルゲームに挑戦

新星出版社

任天堂 ファミリー コンピュータ™
ファミリー ベーシック™

ぼくらのファミコン入門

ファミリーベーシックでオリジナルゲームに挑戦

山下利秋

新星出版社

ぼくらのファミコン入門

新星出版社